WORK BOOK
Paper 2

Practical Problem-solving and Programming

O Levels

Chapter 8

Introduction

2.1.1 Problem-solving and design

Revision Check List (Based on CAIE Syllabus)	
Show understanding that every computer system is made up of sub-systems, which in turn are made up of further sub-systems	
Use top-down design, structure diagrams, flowcharts, pseudo code, library routines and subroutines	
Work out the purpose of a given algorithm	
Explain standard methods of solution	
Suggest and apply suitable test data	
Understand the need for validation and verification checks to be made on input data (validation could include range checks, length checks, type checks and check digits)	
Use trace tables to find the value of variables at each step in an algorithm	
Identify errors in given algorithms and suggest ways of removing these errors	
Produce an algorithm for a given problem (either in the form of pseudo code or flowchart)	
Comment on the effectiveness of a given solution	

System is a set of things working together as parts of a mechanism or an interconnecting network; a complex whole.

System is a set of principles or procedures according to which something is done; an organized sc heme or method.

A **system** is a set of rules, an arrangement of things, or a group of related things that work together to perform a function.

A **system** is **made up** of a number of **subsystems**. Each subsystem can be further divided into subsystems and so on until each sub-system just performs a single action.

For example the human body is made up of the circulatory system, the digestive system, the nervous system and so on.

An automobile has an exhaust system, an electrical system, an ignition system and so on.

A **COMPUTER SYSTEM** is made up of hardware, software & data, communications and people; each computer system can be divided up into a set of sub-systems. Each subsystem can be further divided into sub-systems and so on until each sub-system just performs a single action.

Computer system is often divided up into sub-systems. This division can be shown using top-down design to produce structure diagrams that demonstrate the modular construction of the system. Each sub-system can be developed by a programmer as sub-routine or an existing library routine may be already available for use. How each sub-routine works can be shown by using flowcharts or pseudo code.

- Top-down design
- Structure diagrams
- Flowcharts
- Pseudo code
- Library routines
- Sub-routines

1. Top-Down Design

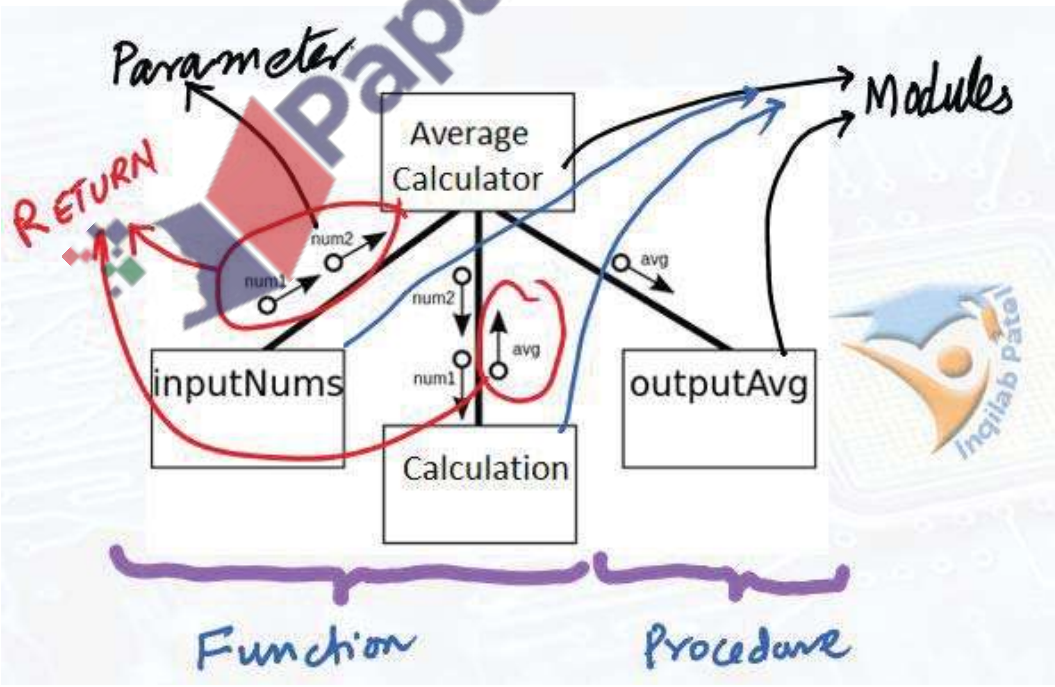
Top-down design is the breaking down of a computer system into a set of subsystems, then breaking each sub-system down into a set of smaller sub-systems, until each sub-system just performs a single action.

This is an effective way of designing a computer system to provide a solution to a problem, since each part of the problem is broken down into smaller more manageable problems. The process of breaking down into smaller sub-systems is called 'stepwise refinement'.

This structured approach works for the development of both large and small computer systems. When large computer systems are being developed this means that several programmers can work independently to develop and test different subsystems for the same system at the same time. This reduces the development and testing time.

2. Structure Diagrams

The **STRUCTURE DIAGRAM** shows the design of a computer system in a hierarchical way, with each level giving a more detailed breakdown of the system into sub-systems.



3. Flowcharts

A **FLOWCHART** shows diagrammatically the steps required for a task (sub-system) and the order that they are to be performed. These steps together with the order are called an **ALGORITHM**. Flowcharts are an effective way to communicate the algorithm that shows how a system or sub-system works.

4. Pseudo code

PSEUDO CODE is a simple method of showing an algorithm, using English-like words and mathematical operators that are set out to look like a program.

5. Library routines

A **LIBRARY ROUTINE** is a set of programming instructions for a given task that is already available for use. It is pre-tested and usually performs a task that is frequently required. For example, the task 'get time' in the checking-for-the-alarm-time algorithm would probably be readily available as a library routine.

6. Sub-routines

A **SUB-ROUTINE** is a set of programming instructions for a given task that forms a subsystem, not the whole system. Sub-routines written in high-level programming languages are called 'procedures' or 'functions' depending on how they are used.

7. Function

A **Function** is a sub-routine that always returns a value.

8. Procedure

A **Procedure** is a sub-routine that doesn't have to returns a value.

Winter 2018 P22

3 Four programming concepts and **four** descriptions are shown.

Draw a line to connect each programming concept to the most appropriate description. [3]

Programming concept	Description
Library routine	A subroutine that does not have to return a value.
Structure diagram	A standard subroutine that is available for immediate use.
Procedure	A subroutine that always returns a value.
Function	An overview of a program or subroutine.

Chapter 9

Algorithm

2.1.2 Algorithm Pseudo code

An algorithm is a series of well-defined steps which gives a procedure for solving a type of problem.
The word algorithm comes from the name of 9th century mathematician al-Khwarizmi (Muhammad Bin Musa Al-Khwarizmi).
In fact, even the word algebra is derived from his book "Hisab al-jebrw'al-muqabala"

2.1.2 Pseudo code
• understand and use pseudo code for assignment, using ←
• understand and use pseudo code, using the following conditional statements:
IF … THEN … ELSE … ENDIF
CASE … OF … OTHERWISE … ENDCASE
• understand and use pseudo code, using the following loop structures:
FOR … TO … NEXT
REPEAT … UNTIL
WHILE … DO … ENDWHILE
• understand and use pseudo code, using the following commands and statements:
INPUT and OUTPUT (e.g. READ and PRINT)
totalling (e.g. Sum ← Sum + Number)
counting (e.g. Count ← Count + 1)
(Candidates are advised to try out solutions to a variety of different problems on a computer using a language of their choice; no particular programming language will be assumed in this syllabus.)

"An **algorithm** is a sequence of steps for a computer program to accomplish a task."

In general, an 'algorithm' is the name given to a defined set of steps used to complete a task.

For instance you could define an algorithm to make a cup of tea. You start by filling the kettle, and then place a tea bag in the cup and so on.

In computer terms, an algorithm describes the set of steps needed to carry out a software task.

This mini-web takes you through the topic of algorithm

Atomic type names

The following keywords are used to designate atomic data types:

1. **INTEGER:**
   A whole number (without fractional part) like COUNT which never requires fractional part
   For example 56, 89, 1

2. **REAL:**
   A number capable of containing a fractional part like Weight may contain fractional Part
   For example 56.8, 89.0, 1.2

3. **CHAR:**
   A single character (may be letter, special character or number but number cannot be used in calculation)
   For example ‘A’, ‘$’, ‘5’

4. **STRING**:
   A sequence of alphanumeric and special characters but number cannot be used in calculation
   For example “Abdullah”, “0300-2724734”, “House No 56 Block 2, PECHS Karachi”

5. **BOOLEAN:** A data type with two possible values
   For example TRUE and FALSE or YES or NO

6. **DATE:** To store a calendar date
   For example 16/04/2010

Literals

Literals of the above data types are written as follows:

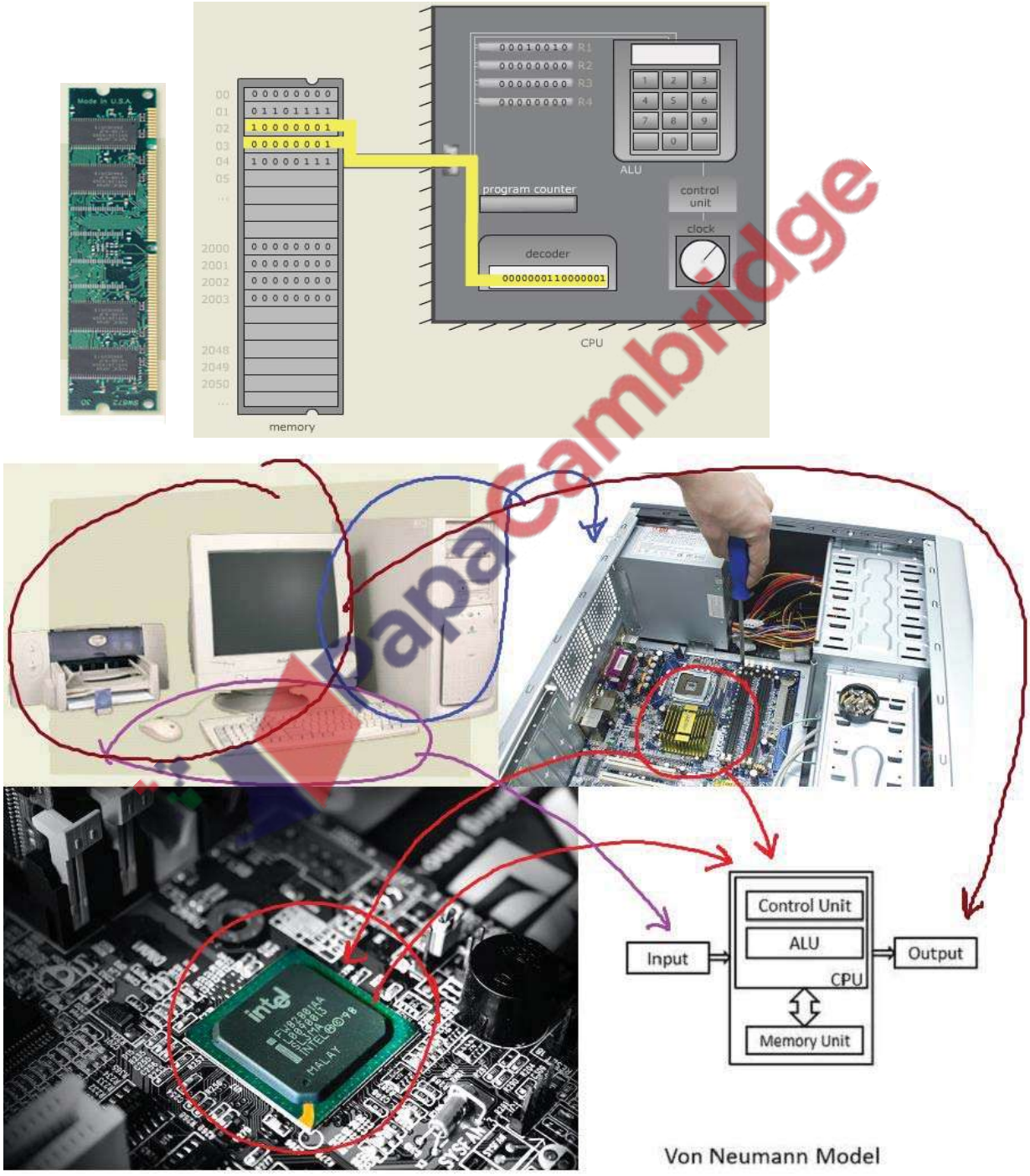
Data Type	Literals
Integers:	Written as normal in the denary system, e.g. 5, -3
Real:	Always written with at least one digit on either side of the decimal point, zeros being added if necessary, e.g. 4.7, 0.3, -4.0, 0.0
Char:	A single character delimited by single quotes, e.g. 'x', 'C', '@'
String:	Delimited by double quotes. A string may contain no characters (i.e. the empty string) e.g. "This is a string", ""
Boolean:	TRUE, FALSE

Variable:

Variable is memory location where a value can be stored. The values stored in a variable are changed during execution.

Identifiers

Identifiers (the names given to variables, constants, procedures and functions) are in mix case. They can only contain letters (A–Z, a–z) and digits (0–9). They must start with a letter and not a digit. Accented letters and other characters, including the underscore, should not be used.



As in programming, it is good practice to use identifier names that describe the variable, procedure

or function they refer to. Single letters may be used where these are conventional (such as i and j when dealing with array indices, or X and Y when dealing with coordinates) as these are made clear by the convention.

Cambridge Ordinary Level
2210 Computer Science June 2019
Principal Examiner Report for Teachers

COMPUTER SCIENCE

Paper 2210/22
Paper 2

Key messages

Candidates must take care when declaring and using variables, constants and arrays as part of a response to ensure that the identifier declared could be used in a program. Identifiers must not contain spaces or other punctuation. Once declared or used the same identifier should be used throughout the answer. Candidates are advised to read through each answer to ensure that no errors have been made.

Cambridge Ordinary Level
2210 Computer Science June 2018
Principal Examiner Report for Teachers

COMPUTER SCIENCE

Paper 2210/21
Paper 2

Key messages

Candidates who had previously completed the tasks for the pre-release (Computer Shop) were able to demonstrate appropriate techniques for solving this problem using a number of valid interpretations of the tasks. These candidates were able to provide answers for ***Section A*** that demonstrated the programs they had written, descriptions of how they had solved tasks and why they had used their chosen methods.

Candidates who were able to explain their code when requested performed better than those who simply wrote out their code.

Candidates should be careful when answering questions pertaining to a specific task in the pre-release materials that their response is related specifically to that task and not generically to the overall pre-release material, or to programming in general. Also, when declaring variables, constants and arrays, it is important that the identifier declared could be used and would work in a program, i.e. it must follow the rules of the programming language to which it relates. Candidates are further advised to ensure that identifiers are descriptive, rather than vague single characters, to demonstrate good programming practice.

Keywords should never be used as variables.

Identifiers should be considered case insensitive, for example, Countdown and Countdown should not be used as separate variables.

Variable declarations

It is good practice to declare variables explicitly in pseudo code.

Declarations are made as follows:

DECLARE<identifier> : <data type>

Example

```
DECLARE Surname : STRING
DECLARE FirstName : STRING
DECLARE DateOfBirth : DATE
DECLARE Section : CHAR
DECLARE Counter : INTEGER
DECLARE TotalToPay : REAL
DECLARE GameOver : BOOLEAN
```

Constant:

Constant is memory location where a value can be stored but the stored value remaining same during execution.

It is good practice to use constants if this makes the pseudo code more readable, as an identifier is more meaningful in many cases than a literal. It also makes the pseudo code easier to update if the value of the constant changes.

Constant declaration

Constants are normally declared at the beginning of a piece of pseudo code (unless it is desirable to restrict the scope of the constant).

Constants are declared by stating the identifier and the literal value in the following format:

CONSTANT<identifier> = <value>

Example

```
CONSTANT HourlyRate = 6.50
CONSTANT DefaultText = “N/A”
```

Only literals can be used as the value of a constant. A variable, another constant or an expression must never be used.

Input and output

Values are input using the INPUT command as follows:

INPUT <identifier>

The identifier should be a variable (that may be an individual element of a data structure such as an array, or a custom data type).

Values are output using the OUTPUT command as follows:

OUTPUT <value(s)>

Several values, separated by commas, can be output using the same command.

Example – INPUT and OUTPUT statements

```
INPUT Answer
OUTPUT Score
OUTPUT "You have ", Lives, " lives left"
```

Note that the syllabus for IGCSE (0478) gives READ and PRINT as examples for INPUT and OUTPUT, respectively.

Arithmetic operations

Standard arithmetic operator symbols are used:

- + Addition
- - Subtraction
- * Multiplication
- / Division

Care should be taken with the division operation: the resulting value should be of data type REAL, even if the operands are integers.

The integer division operators MOD and DIV can be used. However, their use should be explained explicitly and not assumed.

Multiplication and division have higher precedence over addition and subtraction (this is the normal mathematical convention). However, it is good practice to make the order of operations in complex expressions explicit by using parentheses.

Logic operators

The only logic operators (also called relational operators) used are AND, OR and NOT. The operands and results of these operations are always of data type BOOLEAN.

In complex expressions it is advisable to use parentheses to make the order of operations explicit.

Comments

Comments are preceded by two forward slashes // . The comment continues until the end of the line. For multi-line comments, each line is preceded by //.

Normally the comment is on a separate line before, and at the same level of indentation as, the code it refers to. Occasionally, however, a short comment that refers to a single line may be at the end of the line to which it refers.

Example – comments

```
// This is example of comments
// swapping values of X and Y
Temp ← X // temporarily store X
X ← Y
Y ← Temp
```

COUNTING

Counting is used to find how many items are there by incrementing by 1 during each time loop is executed.

It is sometimes necessary to count how many times something happens.

To count up or increment by 1, we can use statements such as:

Count ← Count + 1

(new) (old)

i.e. INCREMENT (old) Count by 1 to get (new) Count

TOTALLING

Totalling is used to calculate running total. We can use a variable such as Total or Sum to hold the running total and assignment statements such as:

Total ← Total + Number

(new) (old)

i.e. ADD Number to (old) Total to obtain (new) Total

Q 1 Summer 2015 P21& 23

5 Explain the difference between a variable and a constant in a program.

.....

.....

.....

.....[2]

Examiner Report Question 5
Well answered by many candidates.

Q 2 Summer 2015 P21& 23

4 Five data types and five data samples are shown below.

Draw a line to link each data type to the correct data sample. [4]

Data type	Data sample
Integer	'a'
Real	2
Char	2.0
String	True
Boolean	"Twelve"

Examiner Report Question 4

Nearly all candidates could link the data type of Boolean with the correct data sample. Some candidates confused Real and Integer data types and/or String and Char data types.

Summer 2016 P21 &P23

3 A program will be written to store information about members of a swimming club.

The following membership details will be recorded:

- Name
- Gender
- Status:
  - Senior
  - Junior
- Fee
- Team member (Yes or No)

(i) Choose a suitable data type for each of the membership details to be recorded. [5]

Membership details	**Data type**
Name	
Gender	
Status	
Fee	
Team member	

Q 4 A marathon runner records their time for a race in hours, minutes and seconds.

An algorithm is shown below in structured English.

INPUT race time as hours, minutes and seconds

CALCULATE race time in seconds

STORE race time in seconds

OUTPUT race time in seconds

The identifier table needs to show the variables required to write a program for this algorithm.

Complete the table.

Identifier	**Data type**	**Description**
RaceHours	INTEGER	The hours part of the race time.

Q 5 A program contains the following code to calculate the circumference of a bicycle wheel, using the wheel size (diameter).

```
CONSTANT Pi = 3.14
INPUT WheelSize
Circumference = Pi * WheelSize
OUTPUT Circumference
```

(a) The code uses one constant and two variables.

(i) State the names of the constant and the variables.

Constant: .....

Variables: .....[2]

(ii) Explain **one** difference between a constant and a variable.

.....

.....

.....[2]

(b) The data type of WheelSize is integer and the data type of Circumference is real number.

Explain the difference between an integer and a real number.

.....

.....

.....

..... [2]

Q 6 Computer programs have to evaluate expressions.

Study the sequence of pseudo code statements.

Write down the value assigned to each variable.

DECLARE h, z, w, r, Perimeter, Area: REAL DECLARE A: BOOLEAN h ← 13.6 w ← 6.4 Perimeter ← (h + w) * 2	Perimeter =..... (1)
r← 10 Area ← 3.14 * (r ^ 2)	Area= (1)
z ← 11 + r / 5 + 3	Z = (1)
A ← NOT (r > 10)	A = (1)

Q 7 Computer programs have to evaluate expressions.

Study the sequence of pseudo code statements.

Give the value assigned to each variable.

The statement may generate an error. If so, write ERROR.

The & operator is used to concatenate strings.

DECLARE N1 : INTEGER	
DECLARE N2 : INTEGER	
DECLARE Answer : REAL	
DECLARE Found : BOOLEAN	
DECLARE IsValid : BOOLEAN	
N1 ← 3	
N2 ← 9	
Answer ← (N1 + N2) / 6	Answer = [1]
Answer ← 3 * (N1 – 2) + N2 / 2	Answer = [1]
IsValid← (N1 > N2) AND (N2 = 9)	IsValid = [1]
Found ← FALSE	
IsValid← (N1 > N2 / 2) OR (Found = FALSE)	IsValid = [1]
Answer ← "1034" & " + " & "65"	Answer = [1]

Q 8 March 2017 P21 (India)

3 There is a program that stores the following data: [8]

• EmployeeID, an employee ID which must be two letters followed by 4 numbers, e.g. TY4587

• Manager, whether the employee is a manager or not

• AnnualHoliday, number of whole days' annual holiday

• PayGrade, the employee's pay grade which must be a single letter A–F

Complete the following table to identify:

• The most appropriate data type for each variable

Variable	**Data type**
EmployeeID	
Manager	
AnnualHoliday	
PayGrade	

Q 9 Winter 2018 P23

2 Describe, giving an example for each, the following data types used in programming.

Integer Description .....

.....

Example .....

String Description .....

.....

Example ..... [4]

Q10 (i) Programming languages support different data types.

Complete the table by giving a suitable data type for each example value. [4]

Example value	Data type
43	
TRUE	
- 273.16	
"- 273.16"	

(ii) Programming languages support different data types.

Complete the table by giving a suitable data type for each example value. [4]

Example value	Data type
"NOT TRUE"	
- 4.5	
NOT FALSE	
132	

(b) Program variables have values as follows:

Programming languages support different data types.

Give an appropriate data type for the following variables from **part (b)**. [5]

Variable	Value	Data type
Married	03/04/1982	
ID	"M1234"	
MiddleInitial	'J'	
Height	5.6	
IsMarried	TRUE	
Children	2	

Summer2019 P21

2 Describe each of the following data types used in programming. In each case, give an example of a piece of data to illustrate your answer. Each example must be different.

Char.....

.....

.....

String.....

.....

.....

Integer .....

.....

.....

Real .....

.....

.....

Date .....

.....

.....

Boolean.....

.....

.....[12]

Q 12 Winter 2019 P22

6 Explain why constants, variables and arrays are used in programming.

Constants .....

.....

.....

.....

Variables .....

.....

.....

..... [2]

The concept of a program

A program is a sequence of instructions or programming language statements written to make a computer perform certain tasks.

Basic Control Constructs:

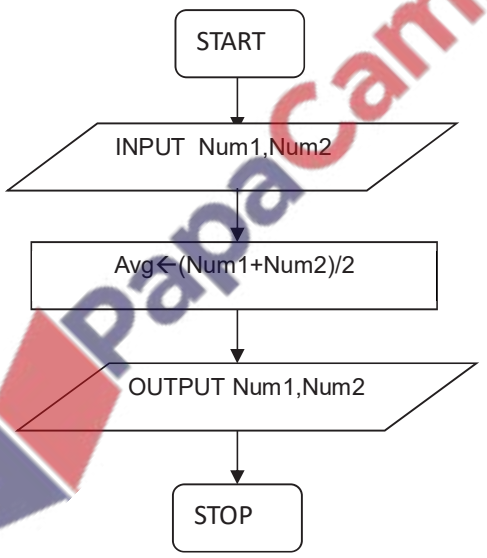
Following are the basic constructs of algorithm and program which controls execution of statements:

1. **Sequence:** One statement is being executed after another in the order they are written
   In following example statement number 'i' will be executed at 1st and then 'ii' then 'iii' and at last statement number 'iv' will be executed:
   i. INPUT Num1
   ii. INPUT Num2
   iii. Total ← Num1 + Num2
   iv. PRINT Total

Flowchart is also drawn in the sequence in which the program is intended to be executed.

Write an algorithm, using flowchart only, which:

- Inputs two numbers
- Calculate their average
- Output average



Problem 1: Input two numbers and output their sum

Problem 2: Input daily wages and number of day worked and output monthly pay.

Q 9.1) Describe the term Computer System and name it's components.

.....

.....

.....

.....

.....

..... [5]

Q 9.2 a) Define the term **algorithm,** name the two ways of representing algorithm.

.....

.....

..... [1]

1. .....

2. ..... [2]

Answer Key: *A series of instructions//sequence of steps;(Designed to) perform a particular task//solve a problem.*

Flowchart and pseudo code

b) Simple algorithms usually consist of three different stages.
Complete the table below. Write each example statement in **program code**.
The second stage has already been given. [5]

Stage	**Example statement**
Process	

Q 9.3) What is top-down design

.....

.....

..... [1]

Q 9.4) Describe following terms and give one example of each

1. Library Routine

.....

.....

..... [1]

2. Sub-routine

.....

.....

..... [1]

3. Function

.....

.....

..... [1]

4. Procedure

.....

.....

..... [1]

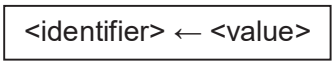
5. Rogue Value

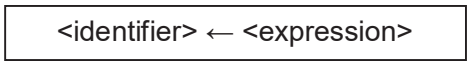
.....

.....

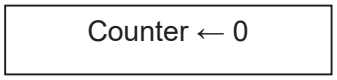
..... [1]

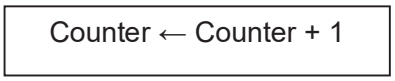
2. **Assignment:** Storing values in a variable is known as assignment.
   The assignment operator is ←.
   Assignments should be made in the following format:
   <identifier> ← <value>

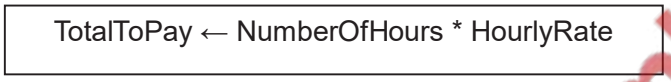




For example:
Counter ← 0
Counter ← Counter + 1
TotalToPay ← NumberOfHours * HourlyRate







3. **Selection (Condition):** Selection determines program flow path on the basis of given condition.

It also decides which statement(s) are to be executed depending upon the result of a given condition. In the following example statement number 'I' will be executed at 1st and then number 'ii'. Execution of statement number 'iii' and 'v' depends upon the result of condition given condition in statement number 'ii':

```
i.   INPUT Marks
ii.  IF Marks>= 50 THEN
iii.         PRINT "Pass"
iv.  ELSE
v.           PRINT "Fail"
vi.  ENDIF
```

4. **Iteration (Loop or Repetition):** Iteration is used to execute a set of instructions multiple times. It is also referred as LOOP or ITERATION.

   In the following example statement number 'ii' will be executed 10 times:

```
i.   FOR Count ← 1 TO 10
ii.        PRINT "Allah is the only God"
iii. NEXT Count
```

A computer's processor can only run a computer program in the form of a file of machine code, which is a sequence of binary codes representing instructions for the processor.

The instruction set for a family of processors is the machine language in which machine code is written for that family of processors.

When machine code runs, the processor repeatedly:

- Fetches an instruction from internal memory
- Decodes the instruction
- Executes the instruction.

Selection:

Selection determines program flow path on the basis of given condition.

Selection decides which statement(s) are to be executed depending upon the result of a given condition.

For selection following statements are used:

- IF
- CASE

IF statements

IF statements are used when there are one or two options.

When there is only one option IF statements without an ELSE clause is written as follows:

```
IF<condition>THEN
        <statements if true>
ENDIF
```

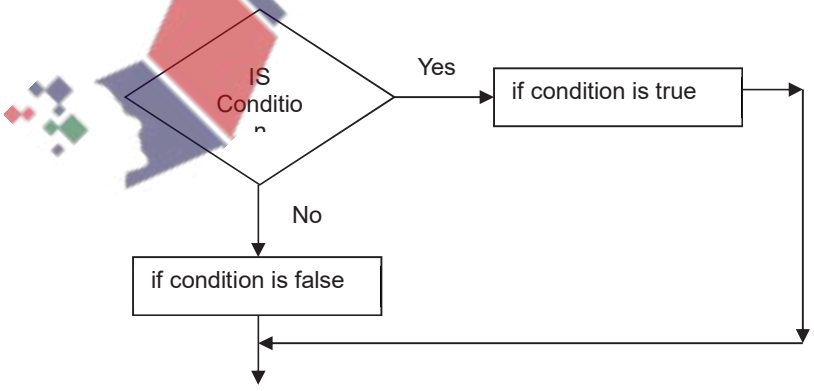
Example

```
IF Number>Largest THEN
   Largest←Number

ENDIF
```

When there are two options IF statements with an ELSE clause is written as follows:

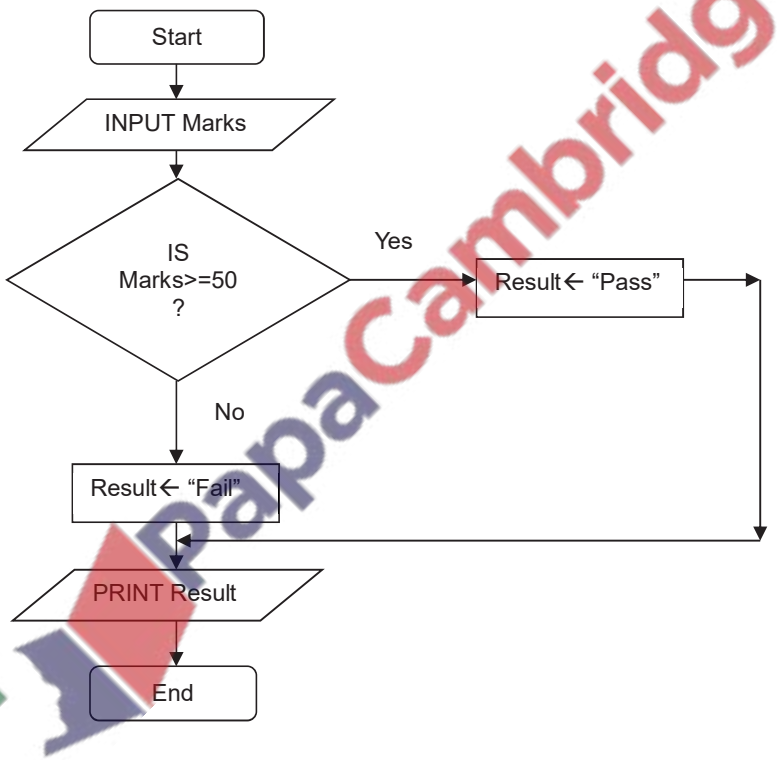
```
IF <condition>THEN
        <statements if true>
  ELSE
        <statements if false>
ENDIF
```



Example

```
IF Marks>=50 THEN
        Result ← “Pass”
 ELSE
        Result ← “Fail”
ENDIF
PRINT Result
```

Note that the THEN and ELSE clauses are only indented by two spaces. (They are, in a sense, a continuation of the IF statement rather than separate statements).
When IF statements are nested, the nesting should continue the indentation of two spaces. In particular, run-on THENIF and ELSE IF lines should be avoided.



CASE statements

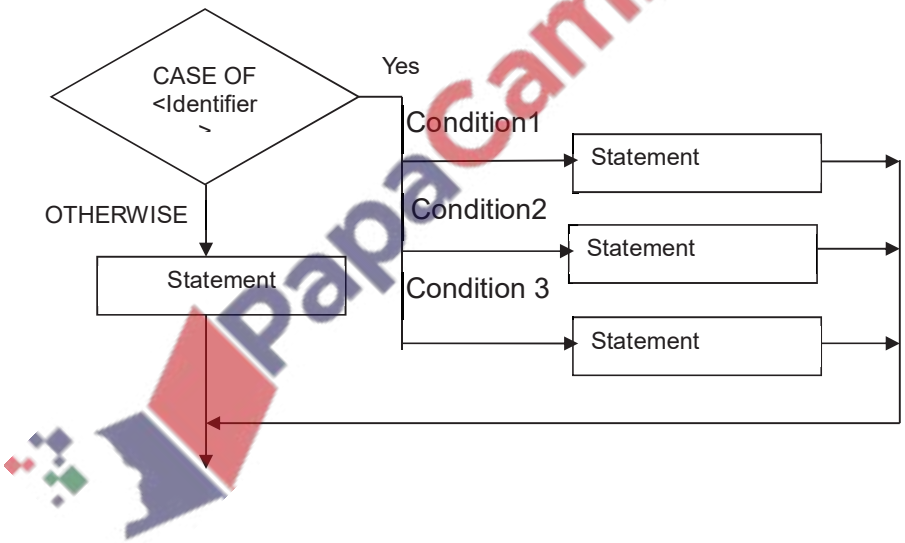
CASE is a conditional statement to deal with many possible outcomes.

CASE statements allow one out of several branches of code to be executed, depending on the value of a variable.
In case selection number of statements are reduced so code become more simplified.
CASE statements are written as follows:

```
CASE OF<identifier>
<value 1> : <statement>
<value 2> : <statement>
  ...
ENDCASE
```

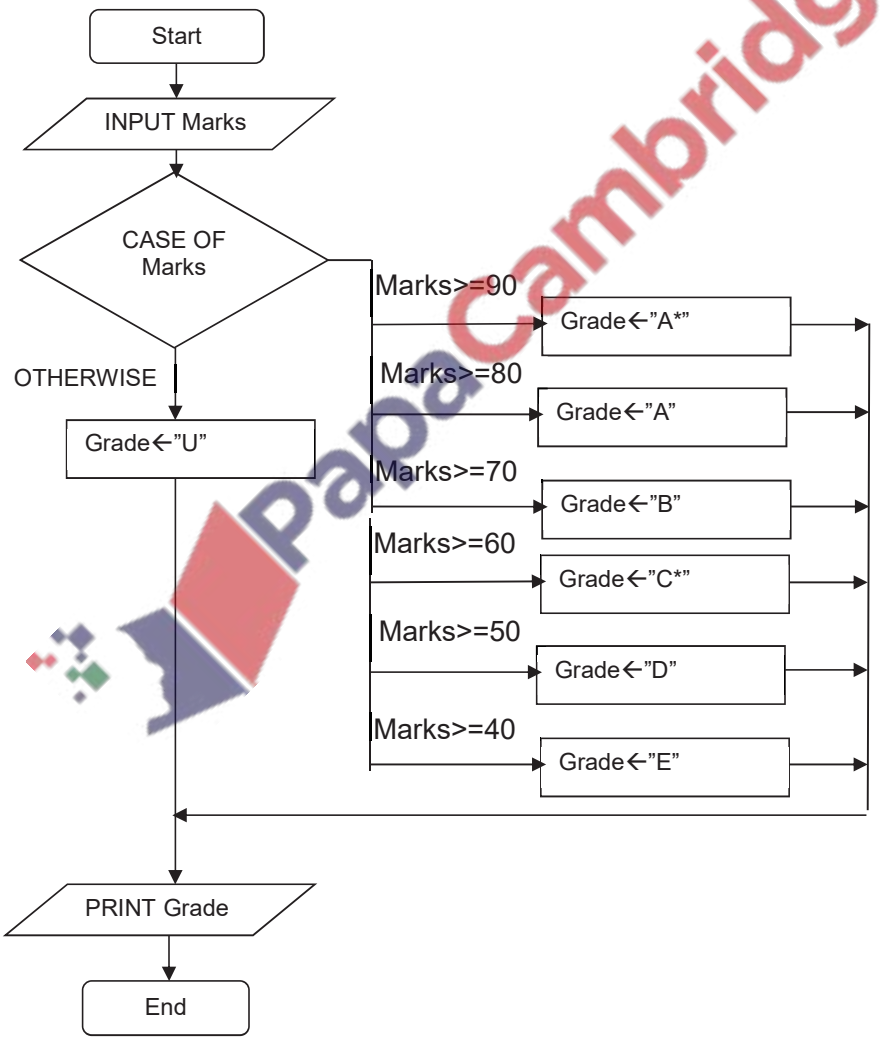
An OTHERWISE clause can be the last case:

```
CASE OF <identifier>
<value 1> : <statement>
<value 2> : <statement>
  ...
  OTHERWISE<statement>
ENDCASE
```



Example – formatted CASE statement

```
INPUT Marks
CASE Marks OF
>=90: Grade ← “A*”
>=80: Grade ← “A”
>=70: Grade ← “B”
>=60: Grade ← “C”
>=50: Grade ← “D”
>=40: Grade ← “E”
OTHERWISE : : Grade ← “U”
ENDCASE
PRINT Grade
```



IF…THEN…ELSE…ENDIF	CASE….OF….OTHERWISE…ENDCASE
Problem: input marks and output result	Problem: input marks and output grade

Problem: input marks and output grade	
IF	CASE

Problem 3: Input marks and output Result, the passing marks is 40 or above.

HW Write algorithm using pseudo code for the following problems:

Problem 4: Input total weight of passengers in a lift output "Overloaded, step out" if Total Weight is above 600 otherwise output "Ready to go".

Problem 5: Input age of candidates for driving license, output "Not allowed to drive" or "Kindly fill in the form". The minimum allowed age for driving is 18 years.

Problem 6: Input age of candidate in an employment center, output "You are not eligible due to age". Allowed age is between 18 and 60 both inclusive.

Problem 7: which inputs price and quantity calculates amount and if billing amount is above 5000 then allows a 5% discount on the billing amount.
Output billing amount, discount and amount after discount

Problem 8) Summer 2018 P22

3 This pseudo code algorithm inputs two non-zero numbers and a sign, and then performs the calculation shown by the sign. An input of zero for the first number terminates the process.

```
INPUT Number1, Number2, Sign

IF Sign = '+' THEN Answer = Number1 + Number2 ENDIF
IF Sign = '-' THEN Answer = Number1 - Number2 ENDIF
IF Sign = '*' THEN Answer = Number1 * Number2 ENDIF
IF Sign = '/' THEN Answer = Number1 / Number2 ENDIF

IF Answer <> 0 THEN OUTPUT Answer ENDIF
```

(a) Complete the trace table for the input data:
5, 7, +, 6, 2, –, 4, 3, *, 7, 8, ?, 0, 0, / [3]

Number1	Number2	Sign	Answer	OUTPUT

(b) Show how you could improve the algorithm written in pseudo code by writing an alternative type of conditional statement in pseudo code. [3]

……
……
……
……
……
……
……
……
……
……

Problem 9) March 2018 P22 (India)

5 Explain the difference between the programming concepts of **sequence** and **selection**. Include an example of a programming statement for each concept in your explanation. [4]

……

……

……

……

……

……

……

……

……

Comments on Question 5

*Candidates found the explanation of the difference between the programming concepts **sequence** and **selection** challenging, with few candidates identifying that programming statements in a sequence were executed one after another whilst selection meant that the path through the program depends on the result of a question. Candidates were more successful in providing suitable examples of programming statements.*

Common errors included confusing sequence or selection with iteration.

Problem 10) Winter 2018 P22

4 A programmer wants to test that the readings from 2000 electricity meters are greater than 400 units and less than 900 units. The programmer uses selection and repetition statements as part of the program. Explain, using programming statements, how selection and repetition could be used in this program.

Selection [2]

……

……

……

……

……

……

Problem 11) Winter 2018 P23

3 Give an example of a pseudo code statement or statements to perform each of the following functions.

A conditional statement [3]

……

……

……

……

Problem 12) Winter 2015 P21 & 22

5 Identify **two** different conditional statements that you can use when writing pseudo code.

1 ……

2 …… [2]

Examiners' Comments Question 5

Many candidates could identify IF as a conditional statement. Candidates with stronger responses throughout also identified CASE.

Problem 13) Summer 2016 P22

6 Identify two different selection statements that you can use when writing pseudo code.

1

2 [2]

Problem 14) Winter 2016 P22

4 IF ... THEN ... ELSE ... ENDIF and CASE ... OF ... OTHERWISE ... ENDCASE

are two different conditional statements that you can use when writing pseudo code.

Explain, using examples, why you would choose to use each conditional statement.

Example 1

Reason for choice

Example 2

Reason for choice [6]

Problem 15) Winter 2017 P22

4 IF … THEN … ELSE … ENDIF is one type of conditional statement used when writing pseudo code.

Identify and describe **another** type of conditional statement that you could use when writing pseudo code. Give a reason why you would use this type of conditional statement.

Conditional statement

Description

Reason [4]

Problem 16) Summer 2018 P21

5 Explain the difference between the programming concepts of **counting** and **totalling**. Include an example of a programming statement for each concept in your explanation. [4]

Problem 17) Summer2019 P21

3 (a) Give an example of a conditional statement using pseudo code. [2]

(b) Describe the purpose of a conditional statement. [2]

Problem 18) Winter 2019 P23

4 The following pseudocode algorithm uses nested IF statements.

```
IF Response = 1    THEN
        X ← X + Y
    ELSE
        IF Response = 2  THEN
                X=X-Y
            ELSE
                IF Response =3  THEN
                        X= X * Y
                        IF Response = 4   THEN
                                X = X/Y
                            ELSE
                                OUTPUT "No response"
                        ENDIF
                ENDIF
        ENDIF
ENDIF
```

(a) Name the type of statement demonstrated by the use of IF … THEN … ELSE … ENDIF [1]

(b) Re-write the pseudo code algorithm using a CASE statement. [4]

Problem 19 (from AS)

The following pseudocode algorithm has been developed to check whether a string contains a valid password.

To be a valid password, a string must:

- be longer than 6 characters
- contain at least one lower case letter
- contain at least one upper case letter
- contain at least one non-alphabetic character.

```
10 FUNCTION Check(InString : STRING) RETURNS BOOLEAN
11
12      DECLARE Index, StrLen, NumUpper, NumLower, NumNonAlpha : INTEGER
16      DECLARE NextChar : CHAR
18      NumUpper ← 0
19      NumLower ← 0
20      NumNonAlpha ← 0
22      StrLen ← LENGTH(InString)
23      IF StrLen < 7
24        THEN
25            RETURN FALSE
26       ELSE
27            FOR Index← 1 TO StrLen
28                  NextChar ← MID(InString, Index, 1)
29                  IF NextChar >= 'a' AND NextChar <= 'z'
30                      THEN
31                          NumLower ←  NumLower + 1
32                  ELSE
33                          IF NextChar > 'A' AND NextChar <= 'Z'
34                              THEN
35                                  NumUpper ← NumUpper + 1
36                          ELSE
37                                  NumNonAlpha ← NumNonAlpha + 1
38                          ENDIF
39                  ENDIF
40            ENDFOR
41      ENDIF
43      IF (NumUpper >= 1) AND (NumLower >= 1) AND (NumNonAlpha >= 1)
44         THEN
45            RETURN TRUE
46      ELSE
47            RETURN FALSE
48      ENDIF50 ENDFUNCTION
```

Rewrite lines 29 to 39 of the original pseudocode using a CASE structure. [4]

Q13 Summer 2019 AS P21

2 (c) The following lines of code are taken from a program in a high-level language.

```
ON x {
    15: Call ProcA
    20: y := 0
    25: y := 99
    NONE: Call ProcError
    }
```

Identify the type of control structure **and** describe the function of the code.

Control structure .....

Description .....

.....

.....

.....[3]

Exercise on Selection

Q1a) Using pseudo code or otherwise, write an algorithm which will input any three numbers and then print the smallest number.

b) Write an algorithm to input three different numbers, and then output the largest number. Use either pseudo code or a flowchart.

Q 2) Write an algorithm, using pseudo code, to input three different numbers, multiply the two larger numbers together and output the result. Use the variables: Number1, Number2 andNumber3 for your numbers and Answer for your result.

Q 4) Write an algorithm, using pseudo code, to input a number between 0 and 100 inclusive. The algorithm should prompt for the input and output an error message if the number is outside this range.

Q 8) Input price and quantity, calculates amount and if billing amount is above 5000 then allows a 5% discount on the billing amount.
Output billing amount, discount and amount after discount

Q 9) Employees of a shop are entitled to a discount of 10% on the value of goods bought from the shop. However if an employee has worked at the shop for five or more years they are entitled to a discount of 20%. Only employees are allowed discounts. The discount on electrical goods is fixed at only 10%.
Using pseudo code or otherwise, write an algorithm which will determine what discount applies when any person buys an item.

Q 10) Customers can withdraw cash from an Automatic Teller Machine (ATM).

• withdrawal is refused if amount entered > current balance

• withdrawal is refused if amount entered > daily limit

• if current balance < $100, then a charge of 2% is made

• if current balance $100, no charge is made

Write an algorithm which inputs a request for a sum of money, decides if a withdrawal can be made and calculates any charges. Appropriate output messages should be included.

Sample algorithm:

```
input amount
    if amount > balance then x = 1                          (2 marks)
        else if amount > daily limit then x = 1             (1 mark)
        else x = 0
        while x = 0
            if balance < 100 then charge = 0.02 * amount    (1 mark)
                else charge = 0
                                                            (1 mark)
        endwhile
if x = 1 then print "Sorry, withdrawal refused"
print charge                                                (1 mark)
```

Marking points

1 mark for checking if amount > balance
1 mark for checking if amount > daily limit
1 mark for some way of testing if withdrawal will be refused (value of x in above)
1 mark for checking if balance < $100…
1 mark …for calculating 2% charge
1 mark for no charge if balance >= $100
2 marks for giving correct outputs [5]

Q 11) A formula for calculating the body mass index (BMI) is:

$$BMI = \frac{\text{weight in kilograms}}{(\text{height in metres}) \times (\text{height in metres})}$$

Using **Flowchart**, write an algorithm that will input weight (kg) and height (m) of students, calculate their body mass index (BMI) and output their BMI and comments on BMI.

BMI <19	Under weight
BMI < =25	Normal Weight
BMI>25	Over weight

Q12) A system uses 5 digit numbers with an additional sixth digit used as a check digit.
(b) Each of the six digits in the number has a digit position. [Total=6]

6	5	4	3	2	1	←Digit position
a	b	c	d	e	f	
					Check digit	

digit in position 1 is the check digit i.e. f
The validity of the check digit is found using the following calculation:

- multiply each digit by its digit position (i.e. ax6, bx5, so on)
- add together the results of the multiplications
- divide the sum by 11
- If the remainder is ZERO then the number is valid

Write an algorithm, using flowchart only, which

- inputs six-digit barcodes in the form a, b, c, d, e and f
- re-calculates the check digit for each number and checks whether the input check digit (e) is correct

Q 13) Summer 2013

A small shop uses barcodes which represent 5 digits. The last digit is used as a check digit.
For example:
a b c d e
0 1 2 3 4
The check digit (e) is found by:
• multiplying the first and third digits (i.e. a and c) by 3
• multiplying the second and fourth digits (i.e. b and d) by 2
• adding these four results together to give a total
• dividing this total by 10
• remainder is check digit (e)

Write an algorithm, using flowchart only, which

- inputs five-digit barcodes in the form a, b, c, d, e
- re-calculates the check digit for each number and checks whether the input check digit (e) is correct

Iteration (Repetition, Loop)

Repetition is used to execute a set of instructions multiple times.
Repetition is also referred as LOOP or ITERATION.
There are following three types of loops:

1. Count-controlled loop
2. Pre-condition loop
3. Post-condition loop

Count-controlled (FOR) loops

Count-controlled loop is used when the number of repetition is already known.
Count-controlled loops are written as follows:

```
FOR <identifier>←<value1> TO <value2>
        <statements>
NEXT<identifier>
```

The identifier must be a variable of data type INTEGER, and the values should be expressions that evaluate to integers.

It is good practice to repeat the identifier after NEXT.

```
FOR <identifier>←<value1> TO <value2> STEP<increment>
        <statements>
NEXT
```

The increment must be an expression that evaluates to an integer. In this case the identifier will be assigned the values from value1 in successive increments of increment until it reaches value2. If it goes past value2, the loop terminates. The increment can be negative.

Example: to input 10 numbers and output their final total

```
Total ← 0
FOR Count ← 1 TO 10
INPUT Number
   Total ←Total + Number
NEXT Count
OUTPUT “The grand total is ”, Total
```

Example: to print 1st 10 even numbers

```
FOR Count ← 1 TO 20STEP 2
      PRINT Count
NEXT Count
```

Pre-condition (WHILE) loops

A loop in which condition is given at the start of loop and which is executed only when the condition is true, is called pre-condition loop.

Pre-condition loops are written as follows:

```
WHILE<condition to repeat> DO
        <statements>
ENDWHILE
```

The condition must be an expression that evaluates to a Boolean.

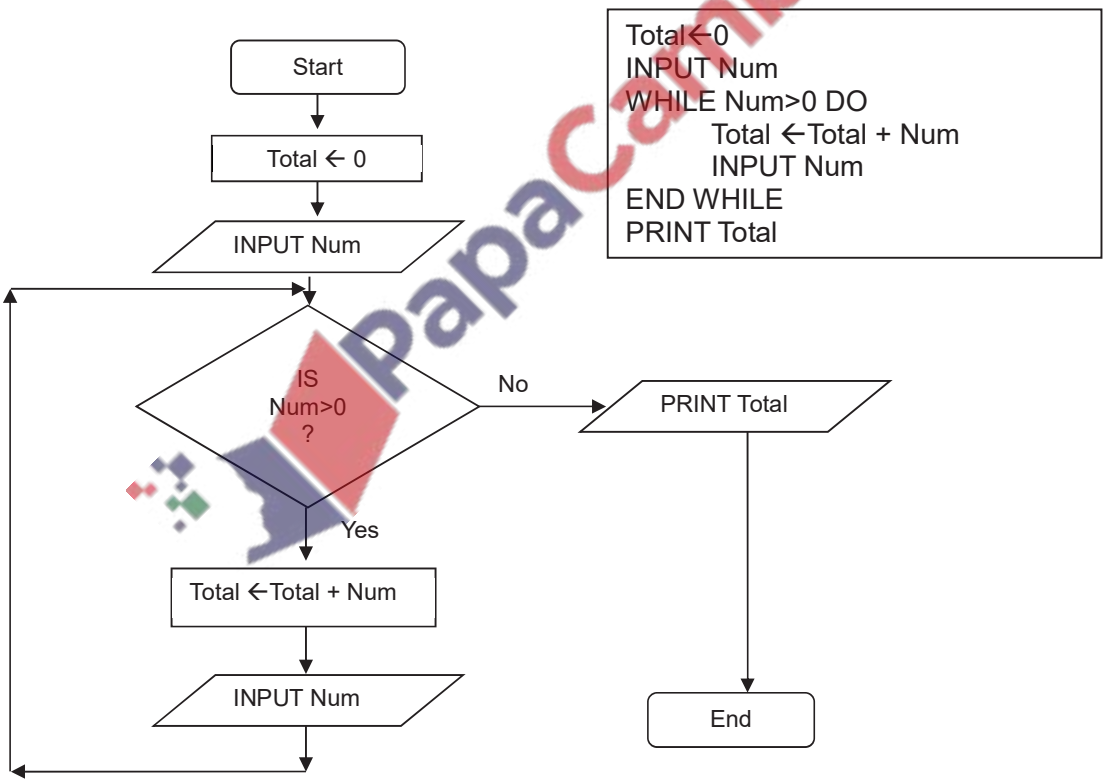
The condition is tested before the statements, and the statements will only be executed if the condition evaluates to TRUE. After the statements have been executed the condition is tested again.

The loop terminates when the condition evaluates to FALSE.

The statements will not be executed if, on the first test, the condition evaluates to FALSE.

Example: To input a series of numbers and calculate total and stops if a –ve number is entered:

The condition is checked at the beginning of the loop. If condition is true loop statements are executed again and again.



Post-condition (REPEAT UNTIL) loops

A loop in which condition is given at the end of loop and which is executed only when the condition is false is called post-condition loop.
It is are written as follows:

```
REPEAT
        <Statements>
UNTIL<condition to stop the loop>
```

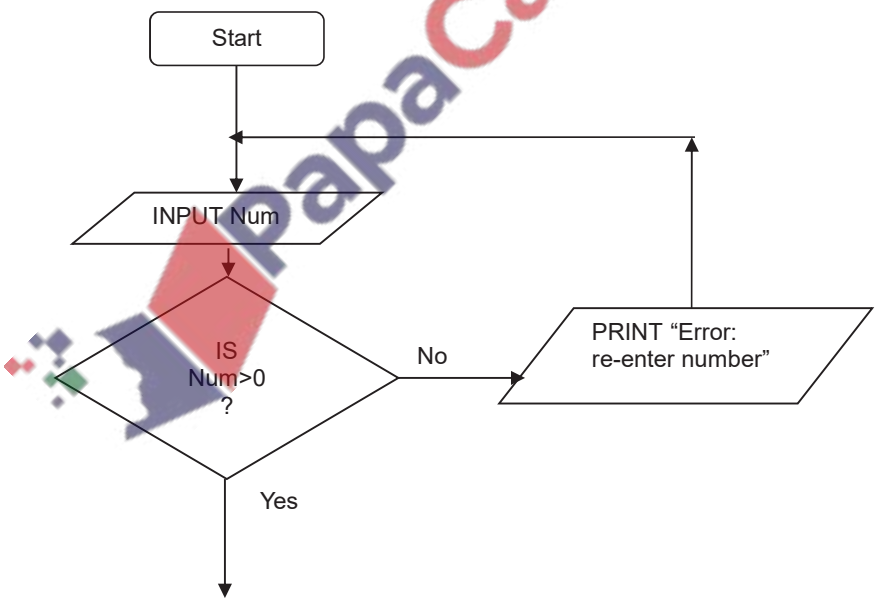
The condition must be an expression that evaluates to a Boolean.

The statements in the loop will be executed at least once. The condition is tested after the statements are executed and if it evaluates to TRUE the loop terminates, otherwise the statements are executed again.

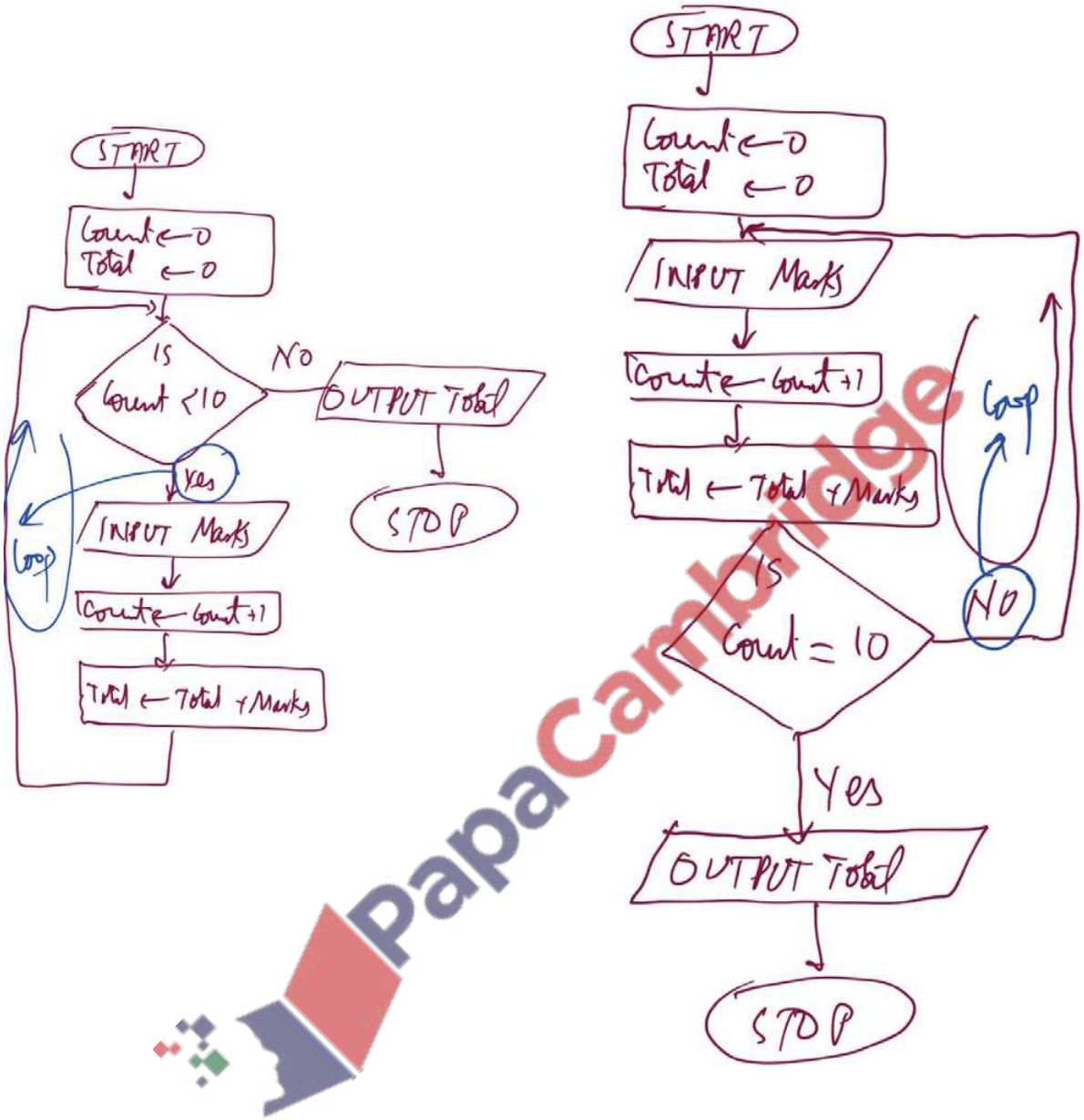
Example: To input and validate a number and to reject it if a negative number is entered and ask to re-enter another number

The condition is checked at the end of the loop. If condition is false loop statements are executed again and again.

```
REPEAT
        INPUT Num
UNTIL Num>0
```



Example: to input 10 numbers and output their final total



Control Construct: Iteration: Iteration is used to execute a set of instructions multiple times. It is also referred as LOOP or ITERATION.

In the following example statement number 'ii' will be executed 10 times:
Problem: Print the name of Allah 10 times.

LOOPING STATEMENTS:

1. FOR … TO…NEXT: Count Controlled loop
2. REPEAT … UNTIL : Post Condition loop
3. WHILE…DO…ENDWHILE: Pre-Condition Loop

Problem: Input daily wages and number of day worked and output monthly pay for 100 employees.

Problem: Input marks of a student in a class output result. Passing marks is 40

Problem: Input marks of 30 students in a class output result of each student. Passing marks is 40

Problem: Print name of Allah 10 times
Flowchart of pre-condition and post condition loops
Problem: Print name of Allah 10 times using all types of loops

Pre-Condition Loop: (WHILE … DO … ENDWHILE)

When condition to continue the loop is given

→

→

→

Problem: To input and add a series of positive numbers in total. Continue this process for input of positive numbers

WHILE	

Post-Condition Loop (REPEAT … UNTIL)

When condition is given at the end of loop

→

→

→

Problem: Input a series of numbers, calculate their total, stop input if total is more than 100

REPEAT … UNTIL Loop	

Differences between

Pre-Condition	Post Condition

Summer 2017 P22

4 An algorithm has been written in pseudo code to input 100 numbers and print out the sum.

A REPEAT … UNTIL loop has been used.

```
Count ← 0
Sum ← 0
REPEAT
    INPUT Number
    Sum ← Sum + Number
    Count ←Count + 1
UNTIL Count > 100
PRINT Sum
```

(a) Find the error in the pseudo code and suggest a correction.

Error 1 .....

Correction .....

.....[2]

(b) Rewrite the correct algorithm using a more suitable loop structure.

.....

.....

.....

.....

.....

.....

..... [3]

There are three different types of looping structures. Write pseudo code for each of following three problems using different looping structure:

a) Input daily temperature for a month of 30 days, calculate and output their total and average.

[6]

Summer 2018 P22

b) Draw a flowchart for an algorithm to input numbers. Reject any numbers that are negative and count how many numbers are positive. When the number zero is input, the process ends and the count of positive numbers is output.

[6]

Extracted from March 2018 P22 (India)

c) Input the weight in kilograms of a passenger stepping into a lift. The lift can take a maximum weight of 640 kilograms. Stop input when total weight is greater than or equal to maximum allowed weight and display message “Ready to Go”

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

Q 9.6) Write an algorithm, using pseudo code and a FOR … TO … NEXT loop structure, to input 1000 numbers into an array.

.....

.....

.....

.....

.....

.....[2]

Summer 2015 P22

(b) Rewrite your algorithm using another loop structure.

.....

.....

.....

.....

.....

.....

.....[4]

Q 9.8a) Write a pseudo code algorithm to input weight of 30 items in kilograms to be loaded on a trailer. Calculate the average weight items under or of 25 kg.

..... [6]

Q 9.8b) Explain how you change your pseudo code to reject any item over 25 Kg.

..... [6]

Control Constructs

Q 1) Write down different statements for following tasks

Input	Output	Selection	Iteration

Q 2) Show what type of programming construct each statement represents.

Complete the table by putting a tick (✓) in the appropriate column for each item.

Item	Statement	Selection	Iteration	Assignment
1	MyScore = 65			
2	FOR IndexVal = 0 TO 99			
3	MyArray[3] = MID(MyString,3,2)			
4	IF MyScore>= 70 THEN			
5	ENDWHILE			
6	ELSE Message = "Error"			

Q 3) Show what type of programming construct each statement represents.

Complete the table by putting a tick (✓) in the appropriate column for each item.

Item	Statement	Selection	Iteration	Assignment
1	WHILE DegF> 37.5			
2	MyName = "Gordon"			
3	DegF = INT(DegF)			
4	ENDIF			
5	CASE OF MyFavourite			
6	UNTIL x = 5			

Summer 2015 P22

4 Four programming concepts and four examples of programming code are shown below. Draw a line to link each programming concept to the correct example of programming code. [4]

Programming concept	Example of programming code
Counting	`Sum = Sum + Value[n]`
Repetition	`IF Value = 10 THEN PRINT 'X'`
Selection	`FOR Counter = 1 TO 10`
Totalling	`Amount = Amount + 1`
	`Sum = Num1 + Num2`

Examiner's comments on Question 4

Nearly all candidates could link the programming concept of selection with the correct example of programming code. Many candidates correctly linked at least three out of the four programming concepts. Due to an issue with this question, a discussion took place between the Principal Examiner and Assessment specialists to consider the impact on candidates in the light of answers seen. No candidates were disadvantaged and the full range of marks was seen.

Summer 2015 P21& 23

6 Identify **three** different loop structures that you can use when writing pseudo code.

1.....

2 .....

3..... [3]

Examiner Report Question 6

Most candidates could identify at least one loop structure. A common wrong answer was to incorrectly identify IF as part of a loop structure.

Summer 2015 P22

5 (a) Write an algorithm, using pseudo code and a FOR … TO … NEXT loop structure, to input 1000 numbers into an array.

.....

.....

.....

.....

.....

.....[2]

(b) Rewrite your algorithm using another loop structure.

.....

.....

.....

.....

.....

.....[4]

Examiner's comments on Question 5

(a) Most candidates attempted the loop structure, better candidates also showed the skill of being able to use the loop counter as the array index. Some candidates misread the question and incorrectly provided program code rather than pseudo code.

(b) Better candidates correctly used REPEAT … UNTIL or WHILE … DO … ENDWHILE structures.

The most challenging aspect was the correct management of the loop counter.

Winter 2015 P21 & 22

5 Identify **two** different conditional statements that you can use when writing pseudo code.

1 .....

2 .....[2]

Examiners' Comments Question 5

Many candidates could identify IF as a conditional statement. Candidates with stronger responses throughout also identified CASE.

Summer 2016 P21 &P23

5 REPEAT ... UNTIL is one type of loop structure.

Identify and describe **two** other types of loop structure that you could use when writing pseudo code.

Loop structure 1: .....

Description: .....

.....

Loop structure 2: .....

Description: .....

.....[4]

Summer 2016 P22

4 Four statement types and four examples are shown below.
Draw a line to connect each statement type to the correct example.

Statement type	Example
Assignment	`FOR X ← 1 TO 10`
Iteration	`READ X`
Input	`PRINT X`
Output	`X ← Y + Z`

[3]

Winter 2016 P21-23

5 REPEAT ... UNTIL and WHILE ... DO ... ENDWHILE are two different loop structures you can use when writing pseudo code.

Explain, using examples, why you would choose to use each type of loop.

Example 1 .....

.....

.....

.....

Reason for choice .....

.....

.....

Example 2 .....

.....

.....

.....

Reason for choice .....

.....

.....[6]

Winter 2016 P22

4 IF ... THEN ... ELSE ... ENDIF and CASE ... OF ... OTHERWISE ... ENDCASE are two different conditional statements that you can use when writing pseudo code.

Explain, using examples, why you would choose to use each conditional statement.

Example 1 .....

.....

.....

.....

.....

.....

Reason for choice .....

.....

.....

Example 2 .....

.....

.....

.....

.....

.....

Reason for choice .....

.....

..... [6]

March 2017 P21 (India)

5 (a) Rewrite the following pseudo code algorithm using a WHILE … DO … ENDWHILE loop.

```
INPUT Num
FOR Counter ← 1 TO 12
        Num ← Num * Counter
        A[Counter] ← Num
NEXT
```

[4]

(b) Explain the differences between a WHILE … DO … ENDWHILE and a REPEAT … UNTIL loop
.

.....

.....

.....

.....

.....

.....

.....

.....[4]

Summer 2017 P22

4 An algorithm has been written in pseudo code to input 100 numbers and print out the sum.
A REPEAT … UNTIL loop has been used.

```
Count ←      0
Sum ←                0
REPEAT
        INPUT Number
        Sum ← Sum + Number
        Count ←Count + 1
UNTIL Count > 100
PRINT Sum
```

(a) Find the error in the pseudo code and suggest a correction.

Error.....

Correction .....

.....[2]

(b) Rewrite the correct algorithm using a more suitable loop structure.

.....

.....

.....

.....

.....

.....

.....[3]

Summer 2017 P21

5 (a) Describe the purpose of each statement in this algorithm.

```
FORI← 1 to 300
        INPUT Name[I]
NEXT I
```

.....

.....

.....

.....

.....[2]

(b) Identify, using pseudo code, another loop structure that the algorithm in part (a) could have used.

.....

.....[1]

(c) Write an algorithm, using pseudo code, to input a number between 0 and 100 inclusive. The algorithm should prompt for the input and output an error message if the number is outside this range.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

Winter 2017 P21

4 (a) Four pseudo code descriptions and **five** pseudo code statements are shown. Draw one line to link each pseudo code description to the correct pseudo code statement. Not all pseudo code statements will be used.[4]

Pseudo code description	Pseudo code statement
A loop that will iterate at least once.	FOR…TO…NEXT
A conditional statement to deal with many possible outcomes.	IF…THEN…ELSE…ENDIF
	WHILE…DO…ENDWHILE
A loop that will iterate a set number of times.	
	CASE…OF…OTHERWISE…ENDCASE
A conditional statement with different outcomes for true and false.	REPEAT…UNTIL

Winter 2017 P22

4 IF … THEN … ELSE … ENDIF is one type of conditional statement used when writing pseudo code.

Identify and describe **another** type of conditional statement that you could use when writing pseudo code. Give a reason why you would use this type of conditional statement.

Conditional statement .....

.....

Description .....

.....

.....

.....

Reason .....

..... [4]

March 2018 P22 (India)

5 Explain the difference between the programming concepts of **sequence** and **selection**. Include an example of a programming statement for each concept in your explanation.

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

Comments on Question 5

*Candidates found the explanation of the difference between the programming concepts **sequence** and **selection** challenging, with few candidates identifying that programming statements in a sequence were executed one after another whilst selection meant that the path through the program depends on the result of a question. Candidates were more successful in providing suitable examples of programming statements.*

Common errors included confusing sequence or selection with iteration.

Summer 2018 P21

5 Explain the difference between the programming concepts of **counting** and **totalling**. Include an example of a programming statement for each concept in your explanation.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Winter 2018 P22

4 A programmer wants to test that the readings from 2000 electricity meters are greater than 400units and less than 900 units. The programmer uses selection and repetition statements as part of the program. Explain, using programming statements, how selection and repetition could be used in this program.

Selection .....

.....

.....

.....

.....

Repetition.....

.....

.....

.....

.....

.....

.....

.....[4]

Winter 2018 P23

3 Give an example of a pseudocode statement or statements to perform each of the following functions.

A condition controlled loop .....

.....

.....

A conditional statement .....

.....

.....

Totalling .....

.....

.....[3]

Winter 2018 P22

4 A programmer wants to test that the readings from 2000 electricity meters are greater than 400units and less than 900 units. The programmer uses selection and repetition statements as part of the program. Explain, using programming statements, how selection and repetition could be used in this program.

Selection .....

.....

.....

.....

Repetition .....

.....

.....

..... [4]

March 2019 P22

4 For each of the **four** groups of statements in the table, place a tick in the correct column to show whether it is an example of **Selection** or **Repetition**. [4]

Statements	Selection	Repetition
FOR X ← 1 TO 10 SUM ← SUM + 1 NEXT X		

WHILE X > 10 DO
  SUM ← SUM + 1
  X ← X - 1
ENDWHILE		

IF X > 10
  THEN
  SUM ← SUM + 1
  X ← X - 1
ENDIF		

REPEAT
  SUM ← SUM + 1
  X ← X - 1
UNTIL X > 10		

Summer2019 P22

4 For each of the **four** groups of statements in the table, place a tick in the correct column to show whether it is an example of **Selection** or **Repetition**. [4]

Statements	Selection	Repetition
FOR A ←1 TO 100 B ←B + 1 NEXT A		
CASE A OF 100: B ←A 200: C ←A ENDCASE		
IF A > 100 THEN B ←A ENDIF		
REPEAT A← B * 10 UNTIL A > 100		

Summer2019 P21

3 (a) Give an example of a conditional statement using pseudocode.

.....

.....

.....

.....

..... [2]

(b) Describe the purpose of a conditional statement.

.....

.....

.....

..... [2]

Summer 2016 P22

2 Read this section of program code that inputs 10 positive numbers and then outputs the total.

```
1 Total = 0
2 Counter = 0
3 REPEAT
4     INPUT Num
5     Total = Total + Num
6     PRINT Total
7     Counter = Counter + 1
8 UNTIL Counter = 10
```

This code works, but it is inefficient.

(i) Suggest **three** improvements that could be made.

1 .....

.....

2 .....

.....

3 .....

.....[3]

(ii) Rewrite the program code with your improvements.

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

Validation and Verification

Validation and verification are two ways to check that the data entered into a computer is correct. Data entered incorrectly is of little use.

Data verification

Verification is performed to ensure that the data entered exactly matches the original source. Verification means checking the input data with the original data to make sure that there have been no transcription errors (transcription means copying the data). The standard way to do this is to input the data twice to the computer system. The computer then checks the two data values (which should be the same) and, if they are different, the computer knows that one of the inputs is wrong. E.g. entering password twice during sig-up. Verification methods include:

At the time of entry	At the time of transmission
Double entry	Parity check
Screen/visual check	Checksum.
	ARQ
	Echo Check

Validation is an automatic computer check to ensure that the data entered is sensible and reasonable. It does not check the accuracy of data.

For example, a secondary school student is likely to be aged between 11 and 16. The computer can be programmed only to accept numbers between 11 and 16. This is a **range** check.

However, this does not guarantee that the number typed in is correct. For example, a student's age might be 14, but if 11 are entered it will be valid but incorrect.

A validation check is a rule that is built into a database to check that the data entered is:

- Sensible
- Reasonable
- Within acceptable boundaries
- Complete

It does NOT mean that the data is actually correct, that requires verification.

There are a number of different validation rules that can be used in a database:

Type Checks - Field data types provide a basic method of validation. Field data types are assigned to fields during the creation of the database table and data types such as Numeric, Boolean, Date/Time and Image restrict what can be entered. If a user tries to enter text in a date field or alphabetic characters in a numeric field, their entry will be rejected.

Range checks - these are used to limit the range of data a user can enter. The 'day' part of a date must be in the range 1 to 31. An exam grade should be in the range 'A'...'G' or 'U'.

Presence checks - these simply check that an entry has been made in a particular field i.e. a null value (empty field) is not permitted. Usually, not every field in a record needs to be filled in, however there are likely to be some that must have a value and the presence check means that the system will not allow the record to be saved until an entry is made. An application for a passport must have the applicant's surname.

Length Checks - All alphanumeric data has a length. A single character has a length of 1 and a string of text such as “Hello World” has a length of 11 (spaces are counted in text strings). A length check ensures that such data is either an exact length or does not exceed a specified number of characters. Mobile phone numbers are stored as text and should be 11 characters in length.

Lookup - A lookup check takes the value entered and compares it against a list of values in a separate table. It can then return confirmation of the value entered or a second list based on the value. One use of lookups restricts users to pre-defined input using drop-down lists. A user is forced to use a list box to select from a predetermined list of valid values.

Check digits - this type of check is used with numbers. An extra 'check digit' is calculated from the numbers to be entered and added to the end. The numbers can then be checked at any stage by recalculating the check digit from the other numbers and seeing if it matches the one entered. One example where a check digit is used is in the 10 digit ISBN number which uniquely identifies books.

The last number of the ISBN is actually the check digit for the other numbers, for example - the ISBN 0192761501.

Following two Methods are used to calculate check digit

Modulo-11 Method:

(i) The position of each digit is first considered:

10	9	8	7	6	5	4	3	2	1 ← Digit Position
0	2	2	1	4	3	2	5	6	? ← Number

(ii) Each digit in the number is then multiplied by it’s digit position and the totals are added together:
i.e. (0x10) + (2x9) + (2x8) + (1x7) + (4x6) + (3x5) + (2x4) + (5x3) + (6x2)
= 0+18+16+7+24+15+8+15+12
= 115 total

(iii) The total is then divided by 11 (modulo 11) and the remainder, if any, is subtracted from 11. The answer then gives the check digit.
i.e. 115/11 = 10 remainder 5
i.e. 11 – 5 = 6 (check digit)
hence, the final number is: 0-221-43256-6

(iv) If Check digit = 10 then it will be represented by X (a representation of 10 in ROMAN numbers)

Modulo-10 Method:

Modulo-10 method is used in check digit calculation in ISBN 13, where the 13th digit of the ISBN code is calculated using the following algorithm.

Steps

1. Find sum of digits at odd position
2. Find sum of digits at even position and multiply result by 3

Example

978-3-12-732320-?

Sum of digits at odd position **9+8+1+7+2+2=29**												
9	**7**	**8**	**3**	**1**	**2**	**7**	**3**	**2**	**3**	**2**	**0**	**?**
Sum of digits at even position x 3 **3 (7+3+2+3+3+)=54**												

3. Add both sums — 29+54=83
4. Find Mod10 — 83 MOD 10=3
5. If r

emainder=0 then

Check digit=0

Else

Check digit=10-Remainder — Check digit 10-3=7

ISBN 978-0747595823

Summary of validation

Validation type	How it works	Example usage
Range check	Checks that a value falls within the specified range	Number of hours worked must be less than 50 and more than 0
Length check	Checks the data isn't too short or too long. Values must be a specific length.	A password which needs to be six letters long
Limit Check	Similar to Range Check but the rule involves only one limit.	>=0 means reject negative numbers. Date of birth must not be later than a date.
Type Check	Checks that the data entered is of a given data type,	Number of brothers or sisters would be an integer (whole number).
Character Check	Checks that when a string of characters is entered it does not contain any invalid characters or symbols,	A name would not contain characters such as %, and a telephone number would only contain digits or (,), and+.
Format Check	Checks the data is in the right format. Values must conform to a specific pattern, for example, two letters followed by six digits followed by a single letter	A National Insurance number is in the form LL 99 99 99 L where L is any letter and 9 is any number
Presence check	Checks that data has been entered into a field	In most databases a key field cannot be left blank
Check digit	The last one in a code are used to check the other digits are correct	Bar code readers in supermarkets use check digits

length check – e.g. only 30 characters in name field
character check – e.g. name doesn't contain numeric chars
range check – e.g. day of month in date is between 1 and 31
format check – e.g. date in the form xx/yy/zz
check digit – e.g. end digit on bar code to check if it is valid
type check – e.g. integer, real
(presence check = 0)

Test Data

Test data is the data that is used in testing of a software system.

In order to test a software application we need to enter some data for testing most of the features. Any such specifically identified data which is used in tests is known as test data.

There are following three types of test data:

- Normal Data
- Abnormal Data
- Extreme Data
- Boundary Data

1. Normal Data

This is the data a computer system should work on.Testing needs to be done to prove that the solution works correctly. In order to do this a set of test data should be used together with the result(s) that are expected from that data. The type of test data used to do this is called **NORMAL DATA**, this should be used to work through the solution to find the actual result(s) and see if these are the same as the expected result(s).

For example, here is a set of normal test data for an algorithm to record the percentage marks from 10 end-of-term examinations for a student and find their average mark:

Normal test data: 50, 50, 50, 50, 50, 50, 50, 50, 50, 50

Expected result: 50

2. Abnormal/Erroneous Data

This is data that should cause the system to tell the user that there is a problem with data entered into the system. Testing also needs to be done to prove that the solution does not give incorrect results. In order to do this, test data should be used that will be rejected as the values are not suitable. This type of test data is called **ERRONEOUS** or **ABNORMAL TESTDATA**; it should be rejected by the solution.

For example erroneous/abnormal data for an algorithm to record the percentage marks from 10 end-of-term examinations for a student and find their average mark could be:

Erroneous/abnormal data:–12, eleven

Expected results: these values should be rejected

3. Extreme Data

When testing algorithms with numerical values, sometimes only a given range of values should be allowed. For example, percentage marks should only be in the range 0 to 100. The algorithm should be tested with **EXTREME DATA**, which, in this case, are the largest and smallest marks that should be accepted. Extreme data are the largest and smallest values that normal data can take.

Extreme data: 0, 100

Expected results: these values should be accepted

4. Boundary Data

This is used to establish where the largest and smallest values occur. For example, for percentage marks in the range 0 to 100, the algorithm should be tested with the following boundary data; at each boundary two values are required, one value is accepted and the other value is rejected.

Boundary data for 0 is –1, 0

Expected results: –1 is rejected, 0 is accepted

Rogue Values

A value that stops input is called Rogue Value.

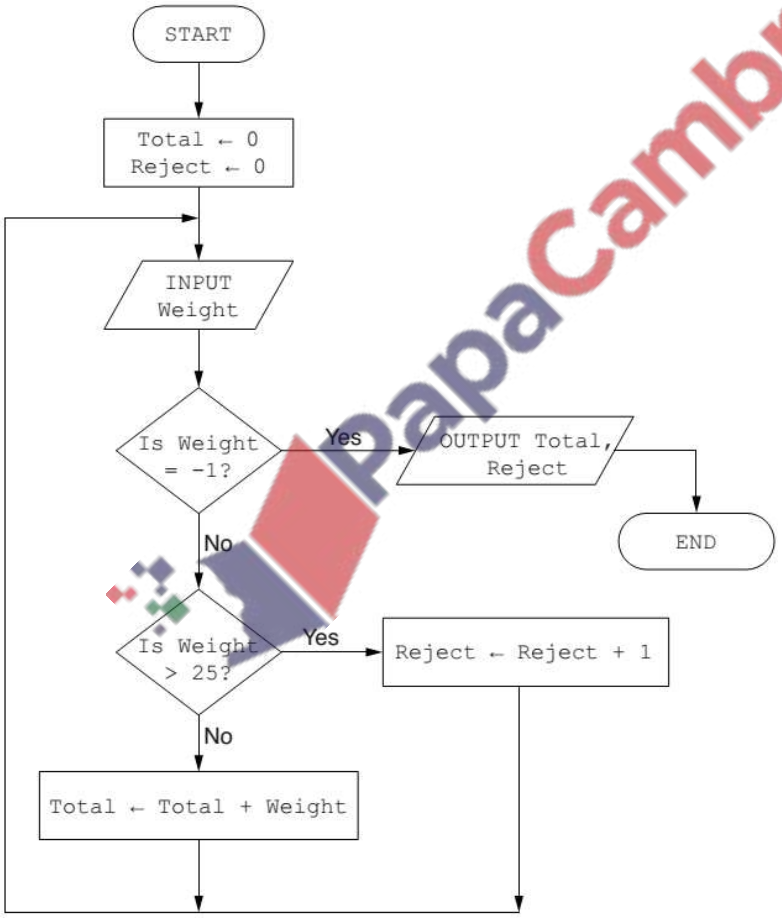
A sequence of inputs may continue until a specific value is input. This value is called a **rogue value** and must be a value which would not normally arise.

A rogue value lets the computer know that a sequence of input values has come to an end.

Example: A number of marks are to be input (terminated by a rogue value of -1). How many of them are over 50?

```
Counter ←0
INPUT Marks
REPEAT
    IF Marks >50 THEN Above50 ←Above50 +1
    INPUT Marks
UNTIL Marks=-1
OUTPUT Above50
```

Example: The flowchart below inputs the weight of a number of parcels in kilograms. Parcels weighing more than 25 kilograms are rejected. **A value of –1 stops the input (a rogue value).**The following information is output: the total weight of the parcels accepted and number of parcels rejected.



Q 8.1) Activity of data validation and verification:

1) What is an automatic computer check to make sure data entered is sensible and reasonable known as?

a) Double entry b) Verification c) Validation

2) What validation type would make sure a post code was entered in the correct format?
a) Length check b) Format Check c) Presence check

3) What validation type would you use to check that numbers fell within a certain range?
a) Range check b) Presence Check c) Check digit

4) What validation type checks that a field is not left blank?
a) Format check b) Length check c) Presence check

5) What validation type uses the last one or two digits to check the other digits are correct?
a) Length check b) Format check c) Check digit

6) What validation type checks a minimum number of characters have been entered?
a) Length check b) Format check

7) Data is to be entered into a computer in the format YYMMDD. Which of the following is not a valid date?
a) 310921 b) 211113 c) 21st June 2004

8) Which of the following statements is false?
a) Validation can check that the data is sensible
b) Validation can check that the data falls between certain allowable boundaries
c) Validation can check that the data is correct

9) Which of the following is NOT a method of verification?
a) Double entry - typing the data in twice and getting the computer to check the second version against the first
b) Using presence, range and length checks to make sure that no mistakes happen
c) Printing out what you have typed in and comparing it against the source data

Write down the name of validation check in front of each description that how it works:

Validation type	How it works
	Checks that a value falls within the specified range
	Checks the data isn't too short or too long. Values must be a specific length.
	Similar to Range Check but the rule involves only one limit.
	Checks that the data entered is of a given data type,
	Checks that when a string of characters is entered it does not contain any invalid characters or symbols,
	Checks the data is in the right format. Values must conform to a specific pattern, for example, two letters followed by six digits followed by a single letter
	Checks that data has been entered into a field
	The last one in a code are used to check the other digits are correct

9.11 What is check Digit

.....

.....

.....[2]

Summer 2012 P12

State two different validation checks and give an example of their use. Each example should be different.

Check 1: ……

Use: ……

Check 2: ……

Use: …… [4]

Q 9.5) Describe Test Data and its four types with the help of examples

Test Data

.....

.....

..... [1]

Test Data Type 1: ……..

..... [1]

Test Data Type 2: ……..

..... [1]

Test Data Type 3: ……..

..... [1]

Test Data Type 4: ……..

..... [1]

Summer 2013 P12

A company requests new customers who register online to give the following details:

• Name

• address

• Type of credit/debit card

• Payment card number

All details must be entered.

(a) (i) Describe one suitable different validation check for each field.

Name: ……

Address: ……

type of credit/debit card: ……

payment card number: …… [4]

Summer 2014 pq11

A hospital holds records of its patients in a database. Four of the fields are:

• date of visit (dd/mm/yyyy)
• patient's height (m)
• 8-digit patient ID
• contact telephone number

The presence check is one possible type of validation check on the data. For each field, give another validation check that can be performed. Give an example of data which would fail your named validation check. A different validation check needs to be given for each field.

field name	**name of validation check**	**example of data which would fail the validation check**
date of visit		
patient's height		
patient ID		
Contact telephone number		

Marking scheme

Field Name	*Name of validation check*	*Example of data which would fail the validation check*
Date of visit	*Format/ type/character check*	*e.g. 2012/12/04e.g. 3rd March 2012*
Patient's height	*range check/limit check*	*can't be < 0 or > 2.5m. e.g. –5, fivee.g. 8, -3,*
Patient ID	*type check* *length check* *range check*	*(can't be < 0 or > 99999999)* *e.g. 3142ABCDe.g. 2131451, 136498207* *e.g. –3, 851341625*
Contact telephone number	*length check, type/character check, format check*	*e.g. 0773141621834e.g. 7H215GD* *e.g. 01223/123456/8901234*

Winter 2015 P23

4 A routine checks the age and height of children who are allowed to enter a play area. The children must be less than 5 years of age and under 1 metre in height.

(a) The first set of test data used is age 3 and height 0.82 metres.

State what type of test data this is.

.....

Give a reason for using this test data.

.....

.....[2]

(b) Provide **two** additional sets of test data. For each, give

• the type of each set of test data

• the reason why it is used

Each type of test data and reason for use must be different.

Set 1 .....

Type .....

Reason .....

.....

Set 2 .....

Type .....

Reason .....

..... [6]

Winter 2015 P21 & 22

4 A routine checks the weight of melons to be sold in a supermarket. Melons weighing under0.5 kilograms are rejected and melons weighing over 2 kilograms are also rejected.

Give an example of each type of test data for this routine

Normal .....

Extreme .....

Abnormal .....[3]

Examiners' Comments Question 4

Most candidates could identify at least one correct example of test data. Examples of normal and abnormal test data were usually correct. Some candidates' examples of extreme test data were incorrect. A correct example of extreme test data would be 0.5 or 2.0.

Winter 2016 P21-23

4 Four validation checks and four descriptions are shown below.
Draw a line to link each validation check to the correct description. [3]

Validation check	Description
Presence check	Numbers between two given values are accepted
Range check	Data is of a particular specified type
Type check	Data contains an exact number of characters
Length check	Ensures that some data have entered

Summer 16 P11, 13

9 Check digits are used to ensure the accuracy of entered data. ()

A 7-digit number has an extra digit on the right, called the check digit.

digit position:	1	2	3	4	5	6	7	8
digit:	–	–	–	–	–	–	–	–

check digit

The check digit is calculated as follows:

•each digit in the number is multiplied by its digit position

•the seven results are then added together

•this total is divided by 11

•the remainder gives the check digit (if the remainder = 10, the check digit is X)

(a) Calculate the check digit for the following number. Show all your working.

4 2 4 1 5 0 8 …

.....

.....

.....

Check digit ..... [2]

(b) An operator has just keyed in the following number:

3 2 4 0 0 4 5 X

Circle below correct if the check digit is correct OR incorrect if the check digit is incorrect.

Correct incorrect

Explain your answer.....

.....

.....

..... [3]

Examiner Report Question 9 (a) and (b)

In part (a) some candidates were able to carry out the first section of the calculation correctly. Some candidates were able to achieve the correct calculation for the final check digit. Candidates need to thoroughly check their calculations. Most incorrect check digits were as a result of addition and division errors and not using the method incorrectly.

In part (b) many candidates were able to identify the check digit was incorrect. Some candidates were then able to explain what the correct check digit would be using the same calculation method. A common mistake that was made was candidates stating the check digit was incorrect because it was a letter.

March 2017 P21 (India)

3 There is a program that stores the following data: [8]

• EmployeeID, an employee ID which must be two letters followed by 4 numbers, e.g. TY4587

• Manager, whether the employee is a manager or not

• AnnualHoliday, number of whole days' annual holiday

• PayGrade, the employee's pay grade which must be a single letter A–F

Complete the following table to identify:

• The most appropriate data type for each variable

• An appropriate validation check for each variable. You must use a different validation check for each variable.

Variable	Data type	Appropriate validation check
EmployeeID		
Manager		
AnnualHoliday		
PayGrade		

Winter 17 P21

3 (a) Explain the difference between a validation check and a verification check.

.....

.....

.....[2]

(b) Describe, using an example, how data could be verified on data entry.

.....

.....

.....[2]

(c) Give two examples of validation of data.

.....

.....

.....[2]

Summer 2017 P22

2 (a) Write an algorithm to input three different numbers, and then output the largest number. Use **either** pseudo code **or** a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

(b) Give **two** sets of test data to use with your algorithm in **part (a)** and explain why you chose each set.

Test data set 1 .....

Reason .....

.....

.....

Test data set 2 .....

Reason .....

.....

.....[4]

Summer 17 P21

4 For each of the **four** statements in the table, place a tick in the correct column to show whether it is an example of **validation** or **verification**. [4]

Statements	Validation	Verification
To automatically check the accuracy of a bar code		
To check if the data input is sensible		
To check if the data input matches the data that has been supplied		
To automatically check that all required data fields have been completed		

March 2018 P22 (India)

A program checks if the weight of a baby is at least 2 kilograms.
Give, with reasons, **two** different values of test data that could be used for the baby's weight.
Each reason must be different.

Value 1 .....

Reason .....

.....

Value 2 .....

Reason .....

.....[4]

Summer 2018 P21

4 A programmer has written a routine to check that prices are below $10.00. These values are used as test data.

10.00 9.99 ten

Explain why each value was chosen.

10.00 .....

.....

.....

9.99 .....

.....

.....

ten .....

.....

.....[3]

Summer 2018 P22

4 A programmer has written a routine to store the name, email address and password of a contributor to a website's discussion group.

(a) The programmer has chosen to verify the name, email address and password.

Explain why verification was chosen and describe how the programmer would verify this data.

.....

.....

.....

.....

.....

.....

.....

.....[4]

(b) The programmer has also decided to validate the email address and the password.

Describe validation checks that could be used.

Email address.....

.....

.....

Password .....

.....

..... [2]

Summer 18 P22

5 A program checks that the weight of a basket of fruit is over 1.00 kilograms and under1.10 kilograms. Weights are recorded to an accuracy of two decimal places and any weight not in this form has already been rejected.

Give **three** weights as test data and for each weight state a reason for choosing it. All your reasons must be different.

Weight 1.....

Reason.....

.....

Weight 2.....

Reason.....

.....

Weight 3.....

Reason.....

..... [6]

Winter 2018 P22

2 (a) Write an algorithm, using pseudo code, to input three different numbers, multiply the two larger numbers together and output the result**.** Use the variables: Number1, Number2 and Number3 for your numbers and Answer for your result.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(b) Give **two** sets of test data to use with your algorithm in **part (a)** and explain why you chose each set.

Test data set 1 .....

Reason .....

.....

Test data set 2 .....

Reason .....

.....[4]

March 2019 P22

5 A programmer restricts input values to less than 90 and greater than 60.

(a) State whether this is called validation or verification.

.....

Name the check that needs to be used.

.....[2]

(b) State **three** different types of test data the programmer would need to use. Give an example of each type and the reason that the programmer chose that test data.

Type 1 .....

Example .....

Reason .....

.....

Type 2 .....

Example .....

Reason .....

.....

Type 3 .....

Example .....

Reason .....

.....[9]

Q 12.83 Summer2019 P22

5 Explain what is meant by **validation** and **verification**.

Give an example for each one.

Validation .....

.....

.....

.....

Example .....

.....

.....

Verification .....

.....

.....

.....

Example .....

.....

..... [6]

Summer2019 P21

4 This section of program code may be used as a validation check.

```
1 PRINT "Input a value between 0 and 100 inclusive"
2 INPUT Value
3 WHILE Value < 0 OR Value > 100
4 PRINT "Invalid value, try again"
5 INPUT Value
6 ENDWHILE
7 PRINT "Accepted: ", Value
```

(a) Give a name for this type of validation check.

..... [1]

(b) Describe what is happening in this validation check.

.....

.....

.....

.....

..... [2]

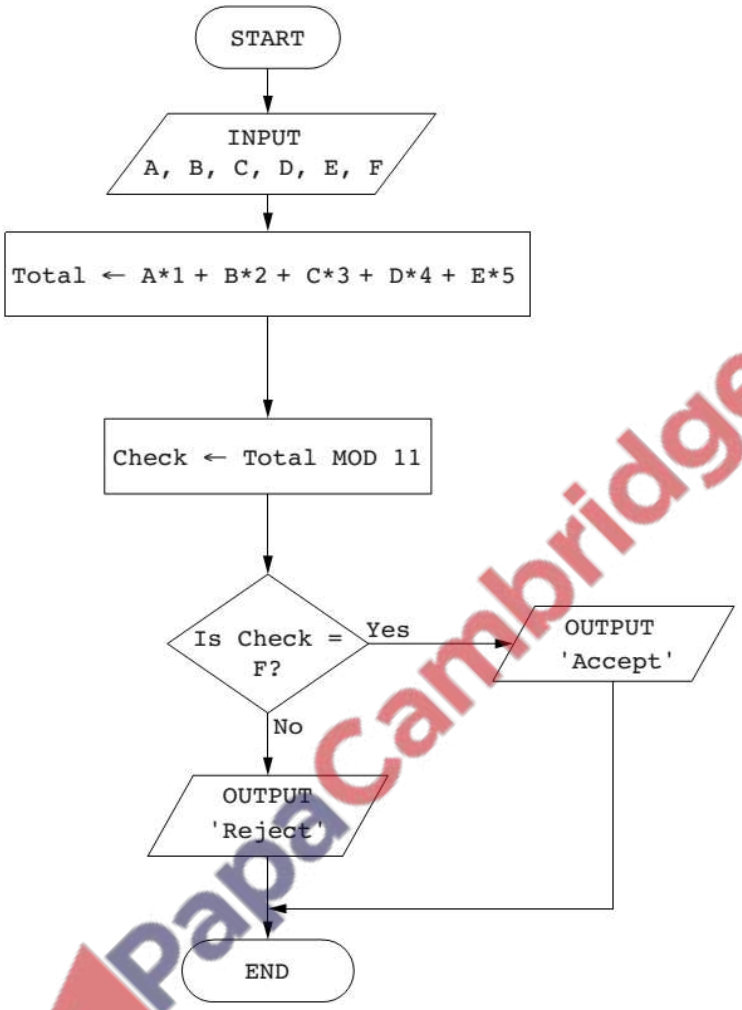
(c) Complete the trace table for this program code using the test data: 200, 300, –1, 50, 60[3]

Value	OUTPUT

(d) Draw a flowchart to represent this section of program code.

Summer 15 P22

3 (a) The flowchart below inputs six single digit numbers. The predefined function MOD gives the value of the remainder, for example, Y ←10 MOD 3 gives the value Y = 1



Complete a trace table for each of the two sets of input data.
Set 1 5, 2, 4, 3, 1, 5
Set 2 3, 2, 1, 0, 7, 3

Trace table set 1: 5, 2, 4, 3, 1, 5

A	B	C	D	E	F	Total	Check	Output

Trace table set 2: 3, 2, 1, 0, 7, 3

A	B	C	D	E	F	Total	Check	Output

Candidate Example response

Example candidate response – high

Complete a trace table for each of the two sets of input data.

Set 1 5, 2, 4, 3, 1, 5

Set 2 3, 2, 1, 0, 7, 3

Trace table set 1 5, 2, 4, 3, 1, 5

A	B	C	D	E	F	Total	Check	Output
5	2	4	3	1	5	38	5	Accept

Trace table set 2 3, 2, 1, 0, 7, 3

A	B	C	D	E	F	Total	Check	Output
3	2	1	0	7	3	45	1	Reject

[4]

Examiner comment – high

The candidate has completed both trace tables correctly.

Total mark awarded = 4 out of 4

Example candidate response – middle

Complete a trace table for each of the two sets of input data.

Set 1 5, 2, 4, 3, 1, 5

Set 2 3, 2, 1, 0, 7, 3

Trace table set 1 5, 2, 4, 3, 1, 5

A	B	C	D	E	F	Total	Check	Output
5	2	4	3	1	5	38	5	Accept
5	4	12	12	5	5			

Trace table set 2 3, 2, 1, 0, 7, 3

A	B	C	D	E	F	Total	Check	Output
3	2	1	0	7	3	45	01	Reject
3	4	3	0	35	3			

[4]

Examiner comment – middle

The candidate has completed A, B, C, D, E and F in both trace tables correctly, the working out to check the calculation against F has incorrectly been included so the marks cannot be awarded. Total, Check and Output are correct for one mark in each trace table. 01 has been accepted as 1.

Total mark awarded = 2 out of 4

Example candidate response – low

Complete a trace table for each of the two sets of input data.

Set 1 5, 2, 4, 3, 1, 5

Set 2 3, 2, 1, 0, 7, 3

Trace table set 1 5, 2, 4, 3, 1, 5

A	B	C	D	E	F	Total	Check	Output
5	2	4	3	1	5			
5	4	12	12	5	5	38	5	Accept
〃	〃	〃	〃	〃	〃	〃	〃	Accept

Trace table set 2 3, 2, 1, 0, 7, 3

A	B	C	D	E	F	Total	Check	Output
3	2	1	0	7	3			
3	4	3	0	35	3	45	1	Reject
〃	〃	〃	〃	〃	〃	〃	〃	Reject

[4]

Examiner comment – low

The candidate has completed A, B, C, D, E and F in both trace tables correctly, the working out to check the calculation against F has incorrectly been included so the marks cannot be awarded. Total and Check are correct, Output is incorrect as both Accept and Reject have been repeated. No marks can be awarded.

Total mark awarded = 0 out of 4

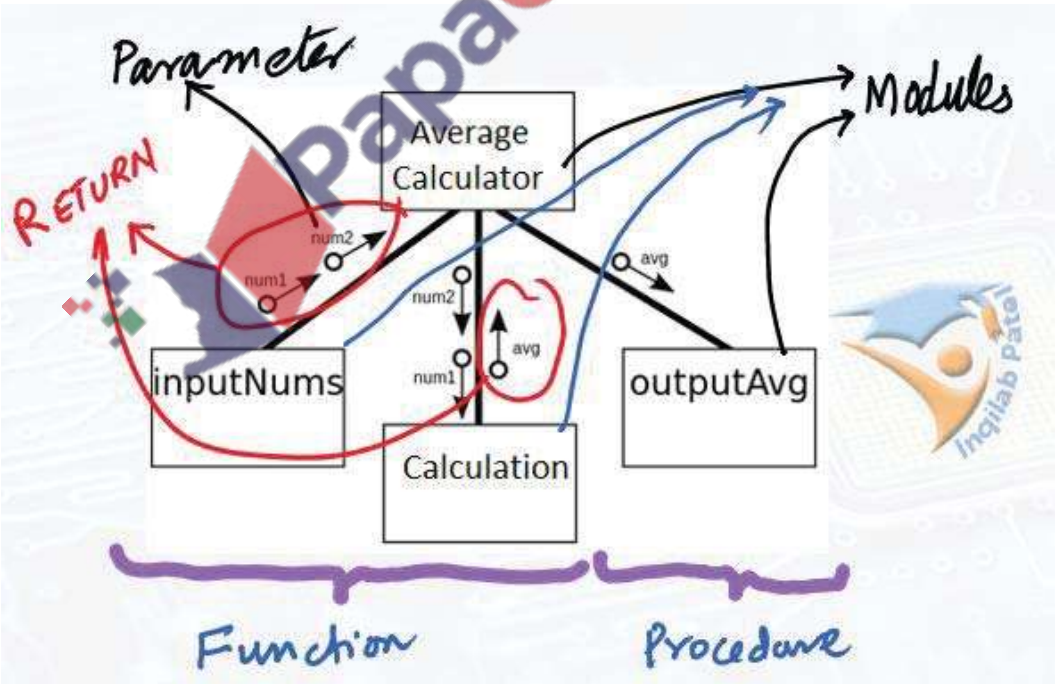
Chapter 10

Structure Chart

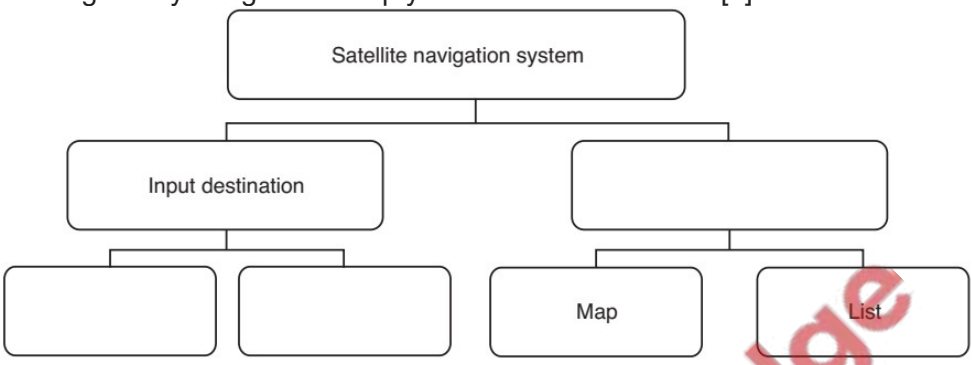
A Structure Chart is a chart which shows the breakdown of a system to its lowest manageable parts. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between modules, showing data transfer between modules using arrows. Structured Charts are an example of a **top-down** design where a problem (the program) is broken into its components. The tree shows the relationship between modules, showing data transfer between the models.

Symbol	Name	Meaning
Module Name	Process	Each Box represents a programming module, this might be something that calculates the average of some figures, or prints out some pay slips
	Data Couple	Data being passed from module to module that needs to be processed.
	Flag	Check data sent to process to stop or start processes. For example when the End of a File that is being read is reached, or a flag to say whether data sent was in the correct format

These individual problems can then be solved and combined according to the links that have been used. If the links between the different blocks are used correctly, the result is a solution to the original problem.



10.1 A satellite navigation system works using destination details entered by the user, either a new destination or chosen from previously saved destinations. The satellite navigation system will then output directions to the destination in the form of either a visual map or a list of directions. A satellite navigation system is an example of a computer system that is made up of sub-systems. This structure diagram shows some of its sub-systems.
Complete the diagram by filling in the empty boxes. [2]



Q 10.2 A modular program design consists of four modules:

Module1 has three sub-tasks. Each sub-task is implemented by a single sub-routine (a function or a procedure).

The subroutine headings are defined as follows:

FUNCTION **Module2** (Weight : REAL) RETURNS BOOLEAN

PROCEDURE **Module3** (Weight : REAL, Customer : STRING, Purchased : DATE)

FUNCTION **Module4** (Purchased : DATE, Account : INTEGER) RETURNS INTEGER

(a) State the term given to values passed between modules.

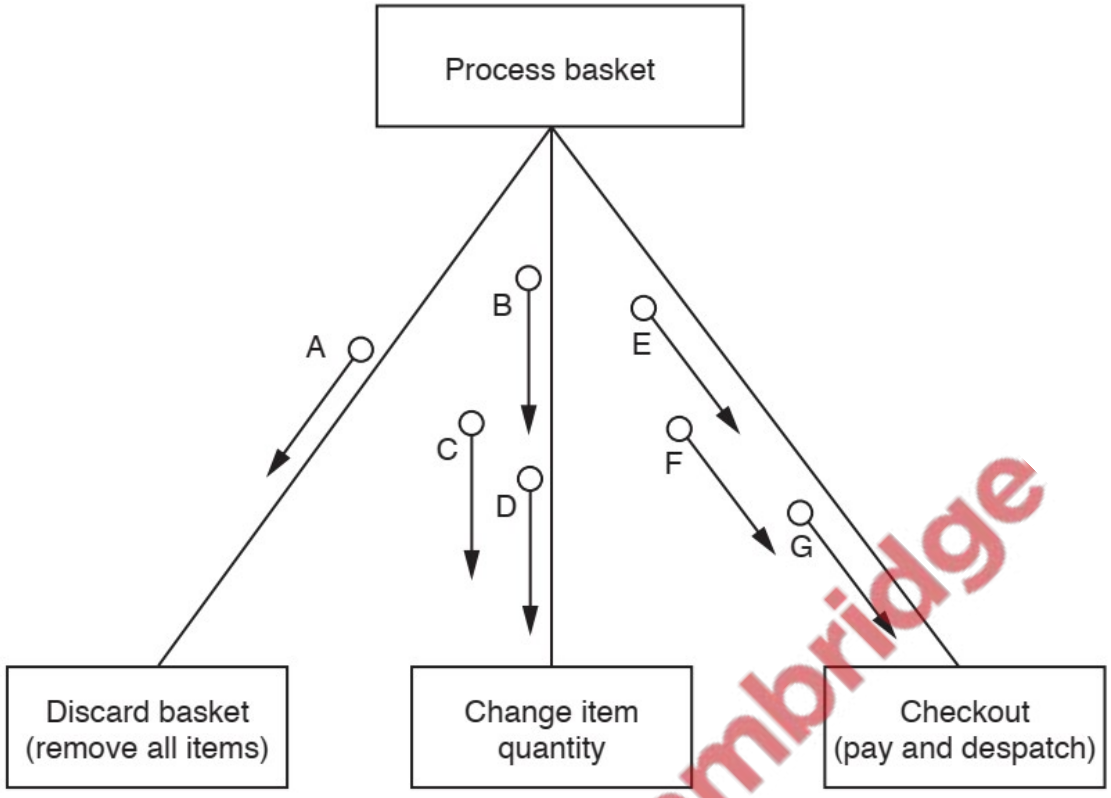
.....[1]

(b) Draw a structure chart to represent the program design.

Use the letters in the table to label the values passed between modules.

Value	Label
Boolean return value	A
Integer return value	B
Account	C
Customer	D
Purchased	E
Weight	F

Q 10.3 The structure chart shows part of the design of a program for an online shopping system.



(a) (i) Draw on the chart to show the following facts.

• Each of the modules at the lower level returns a Boolean parameter, X. [2]

(ii) The parameters A to G shown on the chart will be used to pass the following information.

PaymentDetails
Quantity
BasketID
DeliveryAddress
ItemID

Complete the following table to show the parameter and the information it represents. [3]

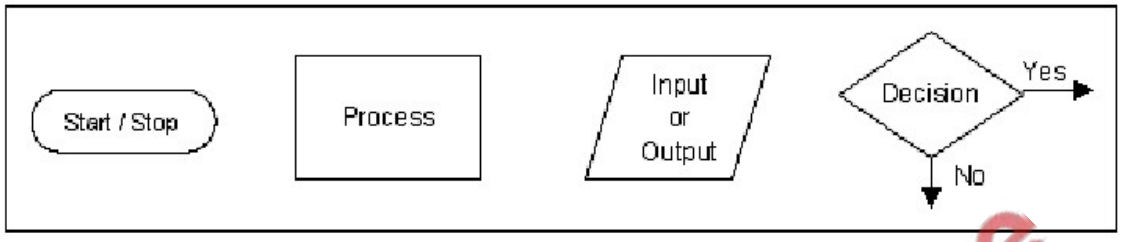
Parameter	Information
A	
B	
C	
D	
E	
F	
G	

Chapter 11

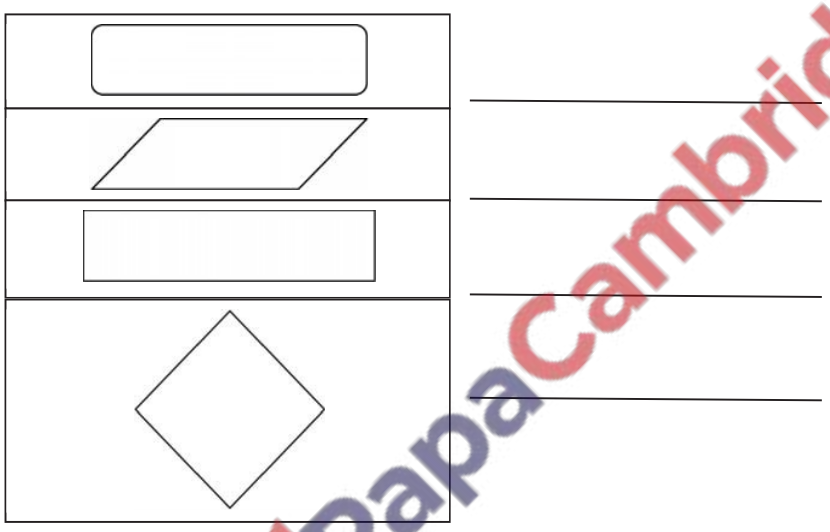
Flowchart

2.1.2 Flowchart

A flowchart is another way of breaking down a program in the form of a diagram.
The following are recognised flowchart symbols:



Write down the name of following flow chart symbols:



Q 11.1) Draw a flowchart that

- Inputs a number
- Find out number is negative or positive
- Output “Positive” or “Negative”

Q 11.2) Draw a flowchart that

- Inputs a number
- Find out number is even or odd (using MOD function)
- Output Even or ODD

Q11.3a) Write an algorithm, using flowchart only, that

- inputs three numbers
- outputs the greatest number

Q 11.3b) Write an algorithm, using flowchart only, that

- inputs three numbers
- outputs the smallest number

Q11.4a) Write an algorithm, using flowchart only, that

- inputs three numbers
- calculate total of two greatest numbers
- Output total

Q 11.4b) Write an algorithm, using flowchart only, that

- inputs three numbers
- calculate total of two smallest numbers
- Output total

COUNTING

Counting is used to find how many items are there by incrementing by 1 during each time loop is executed.

It is sometimes necessary to count how many times something happens.

To count up or increment by 1, we can use statements such as:

Count ← Count + 1
(new) (old)

i.e. INCREMENT (old) Count by 1 to get (new) Count

TOTALLING

Totalling is used to calculate running total. We can use a variable such as Total or Sum to hold the running total and assignment statements such as:

Total ← Total + Number
(new) (old)

i.e. ADD Number to (old) Total to obtain (new) Total

Count-controlled (FOR) loops

Count-controlled loop is used when the number of repetition is already known.

Example Question

a) Draw a flowchart to input 20 numbers and find the total and average of positive numbers

b) Explain how do you change your flowchart to work for 30 numbers that are between 0 and 100.

.....

.....

.....

.....

.....[3]

Redraw the flowchart for part b

Conditional Loop:

A loop which is executed on the basis of a condition.

Pre-condition (WHILE) loop in which condition is given at the start of loop and which is executed only when the condition is true, is called pre-condition loop.

Post-condition (REPEAT UNTIL) loop in which condition is given at the end of loop and which is executed only when the condition is false is called post-condition loop.

Rogue Value A value which stops input, used to terminate loop.

Q 11.8) Draw a flowchart that

Inputs the weight of a number of parcels in kilograms.

Validate parcel (parcels weighing more than 25 kilograms are rejected).

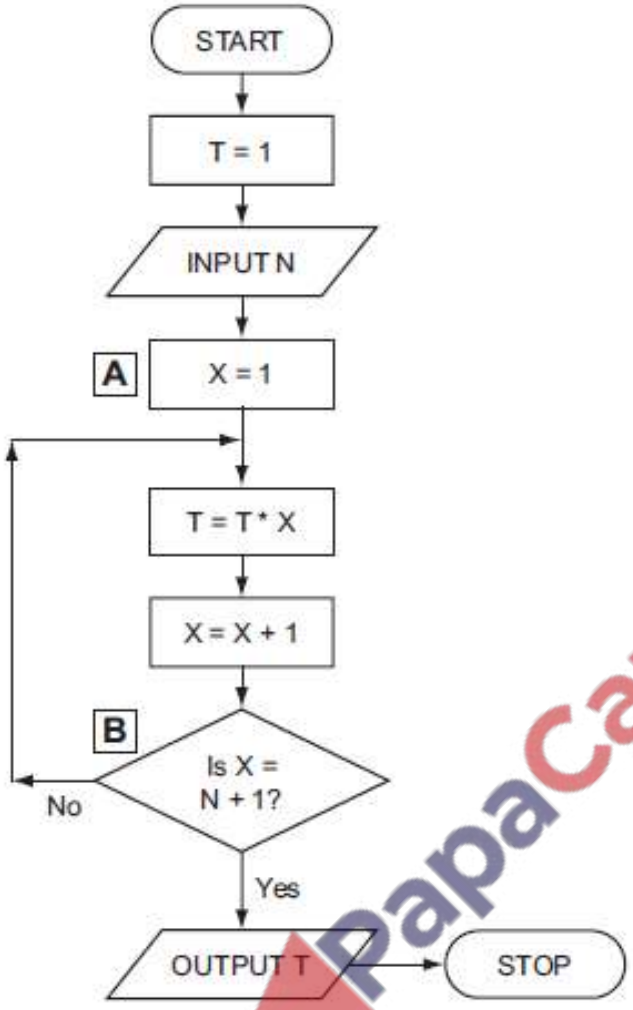
A value of –1 stops the input.

Outputs the total weight of the parcels accepted and number of parcels rejected.

Past Paper Question of same type in Summer 2015 P21& 23 Q 3

Q11.9)Summer 2009

Study the flowchart very carefully.



(a) Complete the table to show what outputs you would expect for the two inputs.[2]

Input N	Output T
5	
1	

(b) Write down a possible LOOP construct for the section A to B in the flowchart using pseudo code.

……

……

…….. [2]

Q 11.10) Draw an algorithm using flowchart that:

- Inputs the height of children who want to ride on a rollercoaster
- Validates height children under 1.2 metres are rejected.
- When eight children have been accepted, outputs message “Ready to go” and number of children rejected.

Past Paper Question of same type in Summer 20162210,0478 P21 &P23

Q11.11a) Draw a flowchart that

- Inputs 50 number
- Find out number is Integer or Real (using INT function)
- Count Integer and Odd Numbers
- Output how many were integer and odd

Q11.11b) Draw a flowchart that

- Inputs a series of numbers
- Calculates their total
- Stops input if a negative number is entered
- Output total.

Q 11.12) Draw a flowchart that

- Inputs temperature for a week (7 days)
- Outputs highest and lowest temperature

Q 11.13) Draw a flowchart that

- Inputs marks of a class of 30 students
- Outputs how many students are pass and how many are fail

Q 11.14) Draw a flowchart that

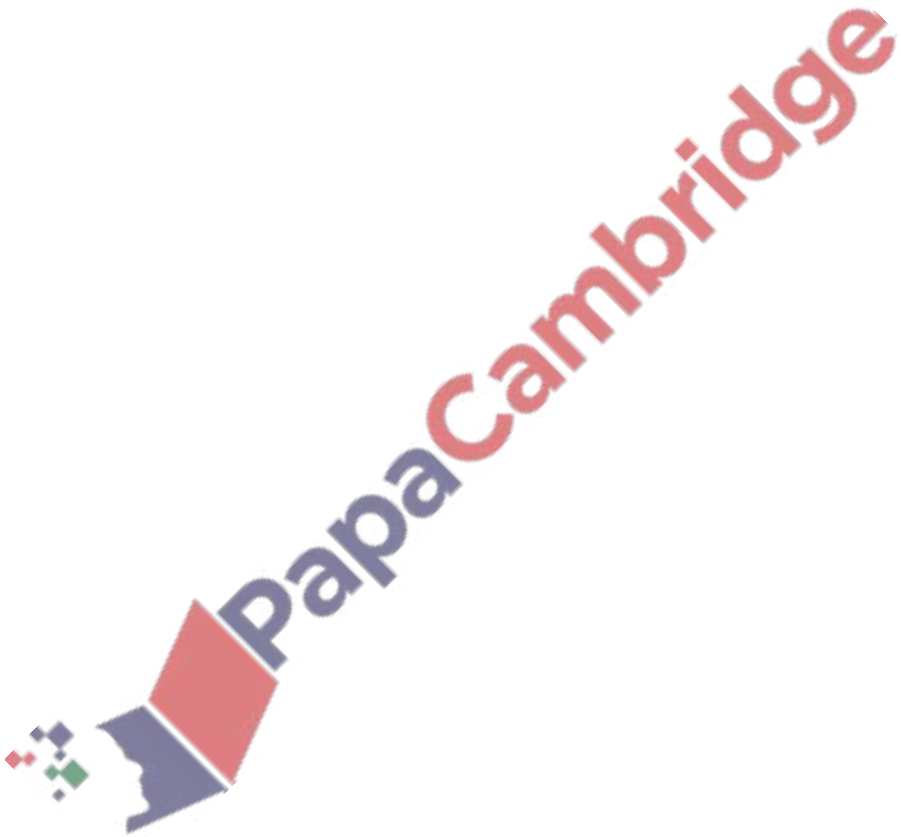
- Inputs per litre price of 5 different brands of milk
- Outputs how average price per litre

Q 11.15) Draw a flowchart that

Inputs a batch of 10 rice sacks for weight

- Validates sacks (sacks should weigh 50 kilograms each. Sacks weighing over 50.5 kilograms or less than 49.5 kilograms are rejected.)
- Outputs number of sacks accepted and the number of sacks rejected.

Past paper flowchart for same type of question in Winter 2017 P22 Q5
March 2018 P22 (India)



Q11.16) Draw a flowchart that

Inputs the weight in kilograms of a passenger stepping into a lift.
The lift can take a maximum of eight passengers or a maximum weight of 640 kilograms.

Q 11.17) Draw a flowchart that

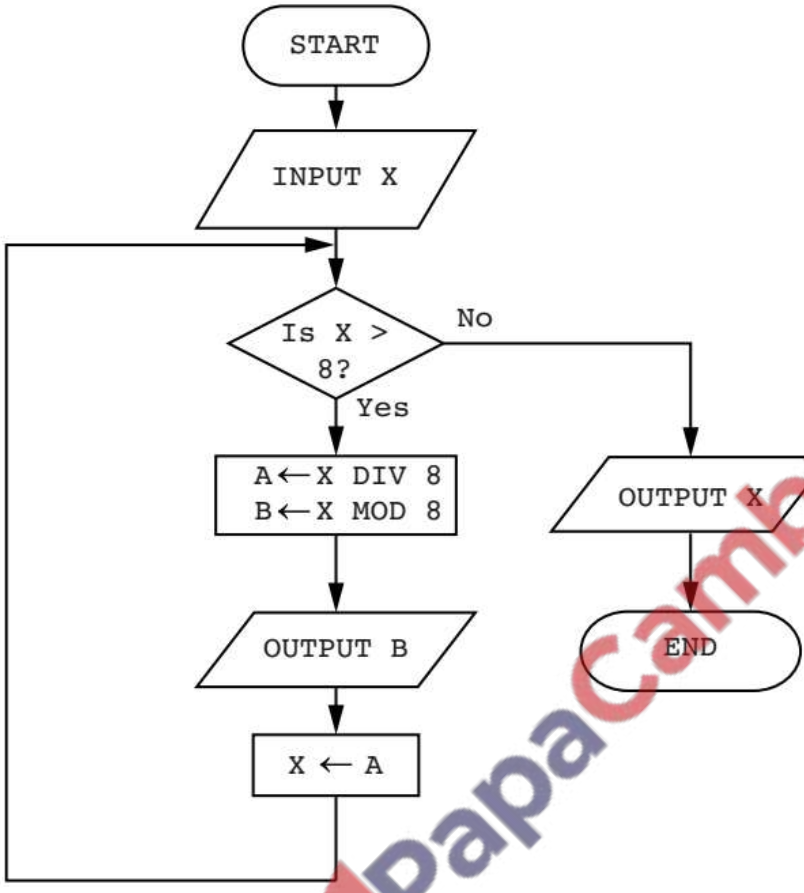
- Inputs name of 10 students in a class and store them in one dimension array
- Display list of names of students

Past paper flowchart for same type of question in Winter 2017 P21 Q5

Q 11.18)Winter 2016 P22 Q 3

Following flowchart is used to convert a denary number into octal (base 8)

The flowchart below inputs an integer. The predefined function DIV gives the value of the division, for example Z ←11 DIV 3 gives the value Z = 3. The predefined function MOD gives the value of the remainder, for example Z ←11 MOD 3 gives the value Z = 2.



Complete a trace table for each of the two input values **33** and **75**. [4]

Trace table for input value **33**

X	A	B	OUTPUT

Trace table for input value **75**

X	A	B	OUTPUT

Q 11.19) Draw a flowchart to convert denary number into binary

Test your flowchart by completing following trace table to convert denary number 20 into binary [4]

X	A	B	OUTPUT

Q 11.20)Winter 2015 P23 Q 3

(a) This pseudo code inputs an integer. The predefined function DIV gives the value of the division, e.g. Y 10 DIV 3 gives the value Y = 3. The predefined function MOD gives the value of the remainder, e.g. Y 10 MOD 3 gives the value Y = 1.gives the value Y = 1.

```
INPUT X
WHILE X > 15
  DO
  T1 ← X DIV 16
  T2 ← X MOD 16
  CASE T2 OF
    10:OUTPUT A
    11:OUTPUT B
    12:OUTPUT C
    13:OUTPUT D
    14:OUTPUT E
    15:OUTPUT F
    OTHERWISE OUTPUT T2
  ENDCASE
  X ← T1
ENDWHILE
CASE X OF
  10:OUTPUT A
  11:OUTPUT B
  12:OUTPUT C
  13:OUTPUT D
  14:OUTPUT E
  15:OUTPUT F
  OTHERWISE OUTPUT X
ENDCASE
```

Complete a trace table for each of the **two** input values 37 and 191.

Trace table for input value 37

X	T1	T2	OUTPUT

Trace table for input value 191

X	T1	T2	OUTPUT

(b) State the purpose of the pseudo code in **part (a)**.

.....

.....[2]

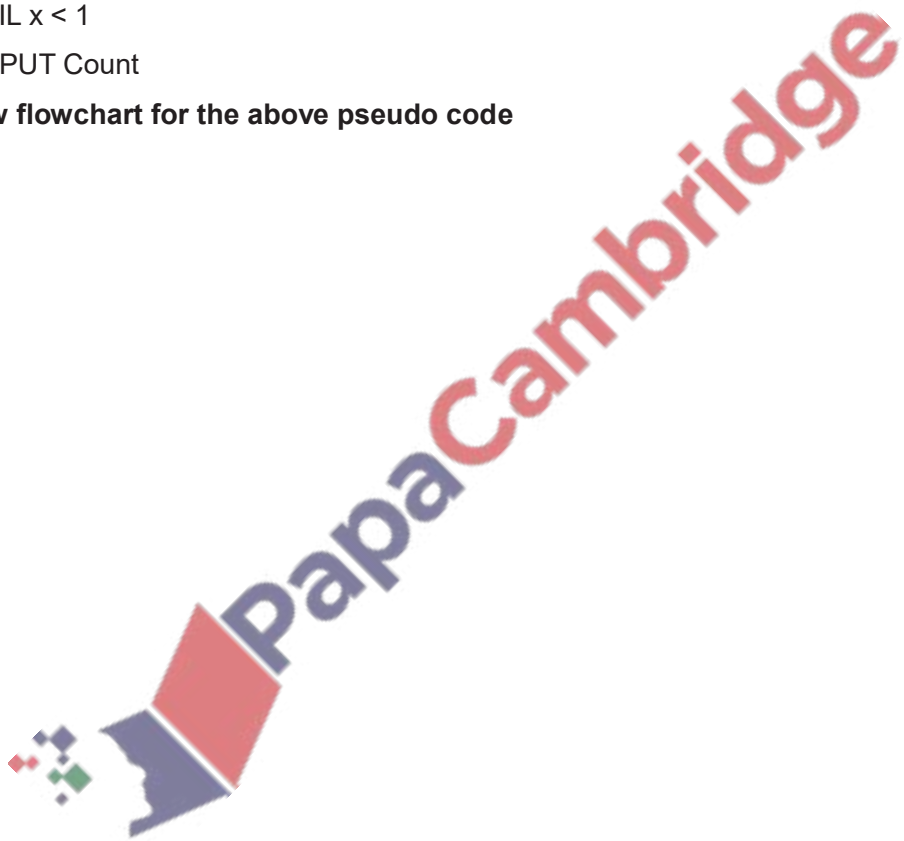
Draw flowchart for the above pseudo code

Past paper question of same type in Winter 2015 P21 & 22

Q 11.20) Following flowchart is used to count digits in a number

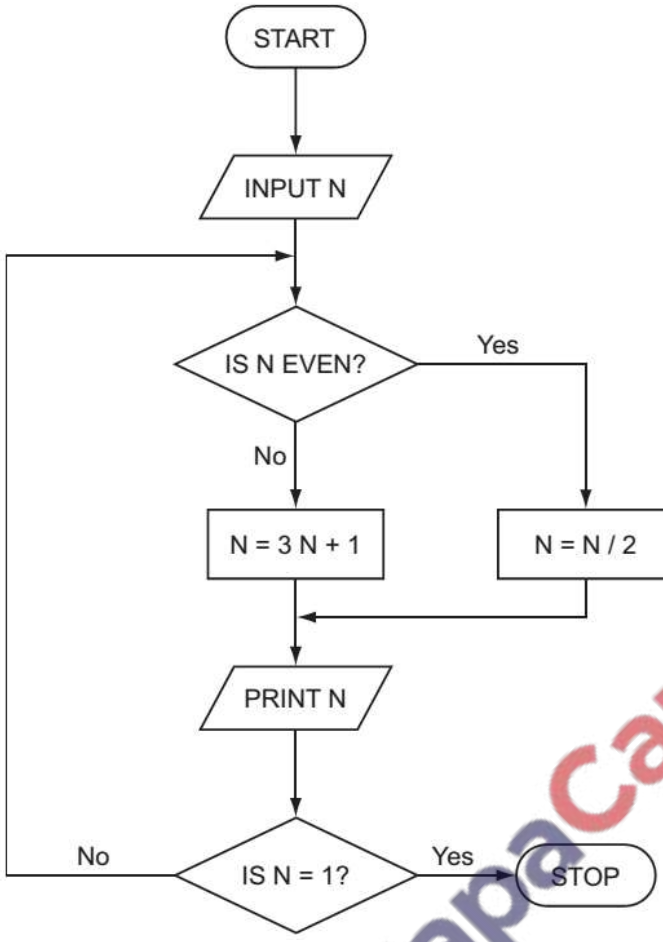
```
DECLARE Count: Integer
DECLARE Num, x: Real
Count ← 0
INPUT Num
x ← PIN
REPEAT
    x ← x/10
    Count ← Count + 1
UNTIL x < 1
OUTPUT Count
```

Draw flowchart for the above pseudo code



Finding Output from flowchart

Q 11.21)Summer 2006



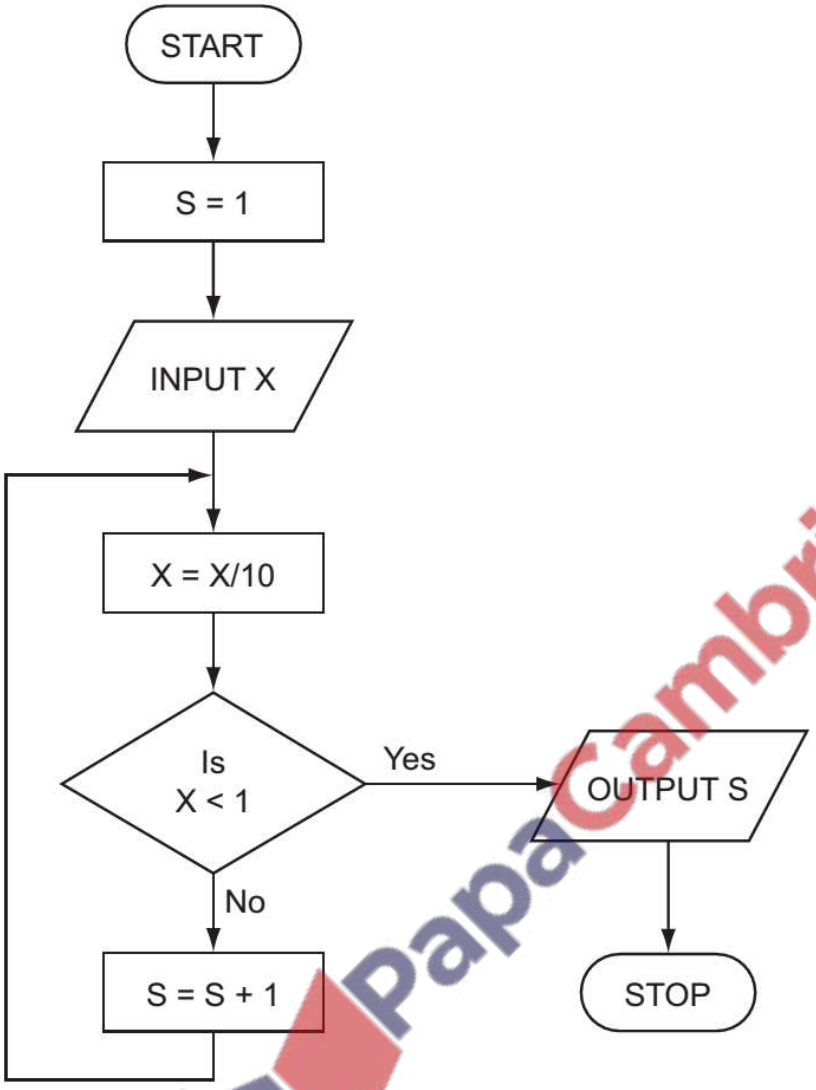
Trace the flow chart using the numbers 2 and 3. Write down each of the values of N in the order that they are printed out.

(a) 2 .....[1]

(b) 3 .....[2]

Q11.22)Summer 2007

Study the following flowchart very carefully.



(a) Complete the following table showing the expected output from the flowchart for the three sets of input data: [3]

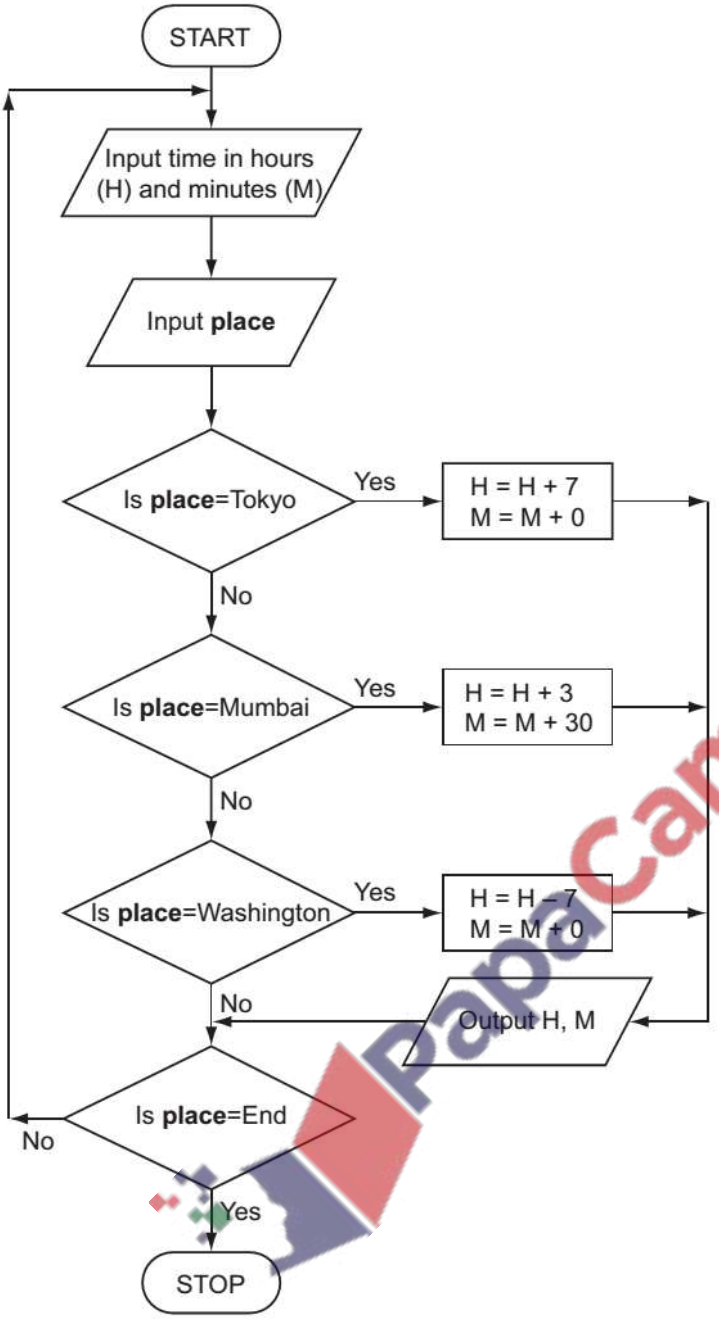
INPUT X	OUTPUT S
48	
9170	
- 800	

(b) Input data needs to go through a validation process.
(i) Explain the term validation.
……

(c) (ii) Describe one type of validation check
…… [2]

Q 11.23) Winter 2007

Majid lives in Cairo but often travels to Tokyo, Mumbai and Washington. A flow chart has been written so he can work out the local time in these three places.



(a) What output would be produced from the following input? [2]

Input			Output	
place	hours (H)	minutes (M)	H	M
Tokyo	11	15		
Mumbai	15	10		

(b) What problem would occur if place = Mumbai and H = 15 and M = 30?

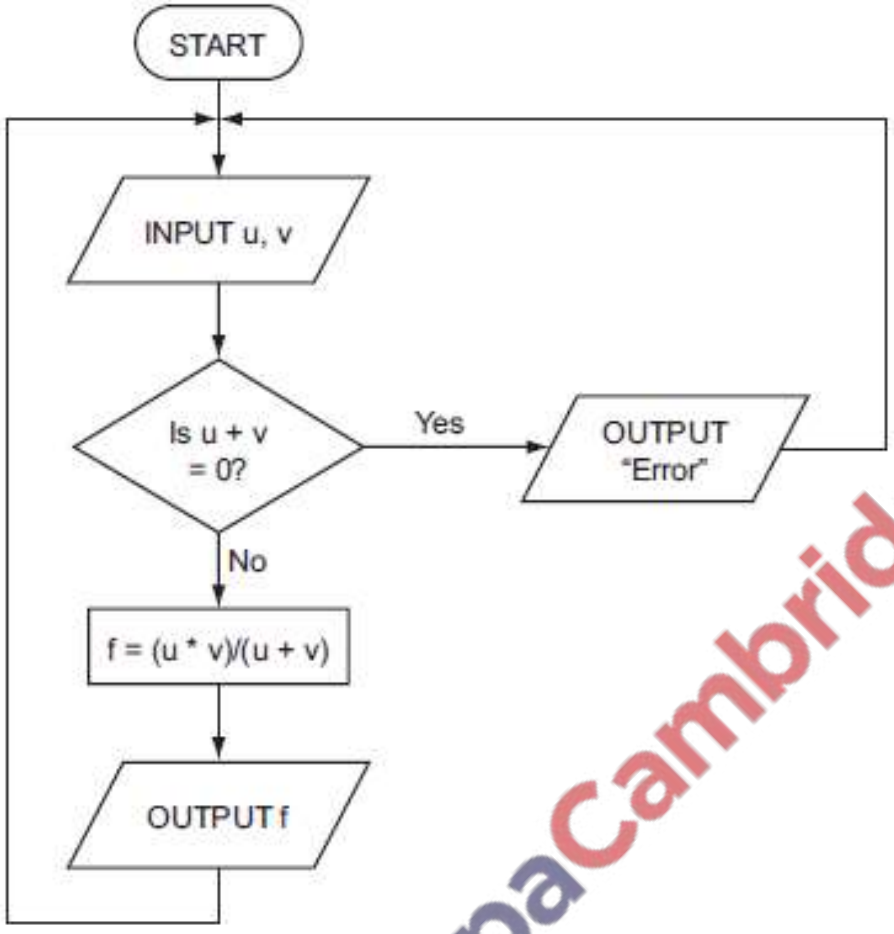
…… [1]

(c) What problem would occur if place = Washington and H = 4 and M = 0?

…… [1]

Q 11.24) Summer 2008

The following flowchart inputs two numbers, carries out a calculation and then outputs the result.



(a) Complete the following table for the three sets of input data. [3]

INPUT		OUTPUT
U	V	
5	5	
6	-6	
12	4	

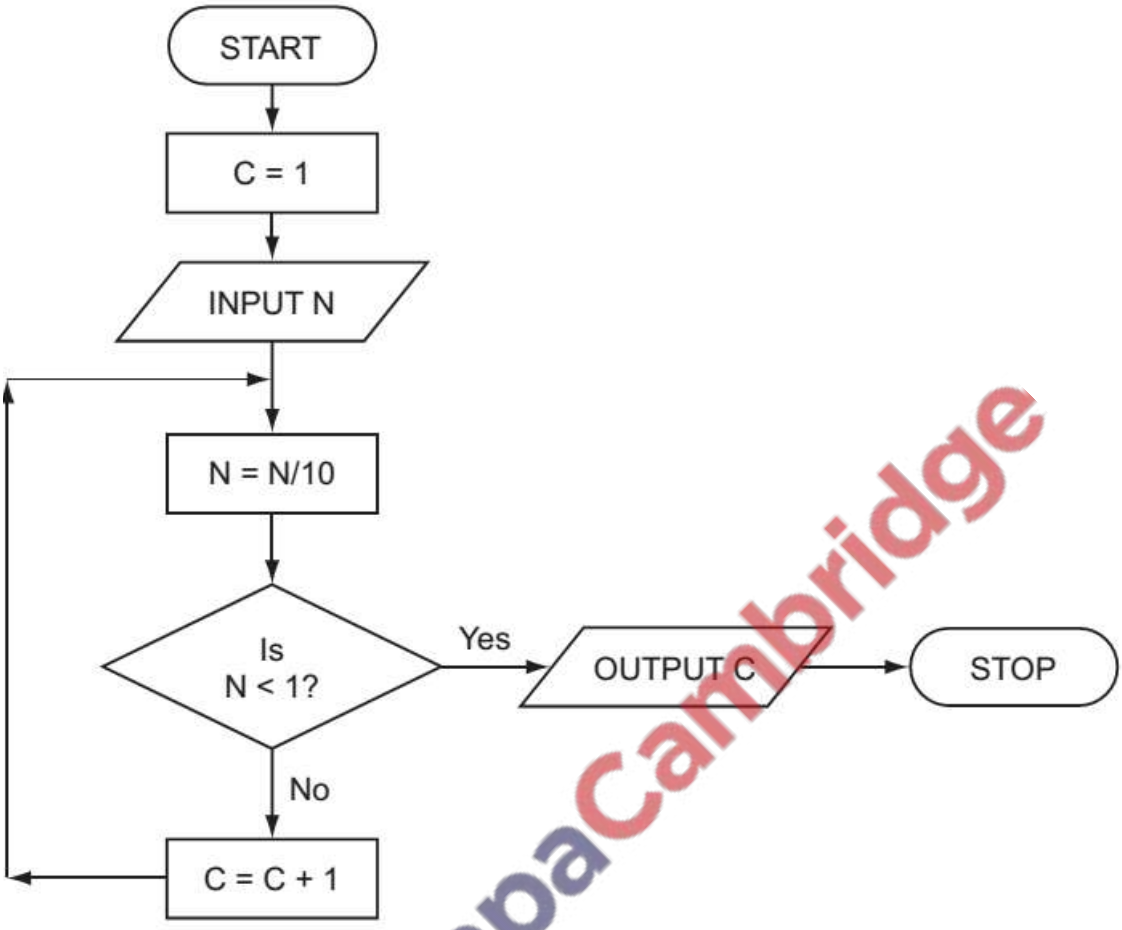
(b) The above algorithm has been placed in a library of routines. Give one advantage of doing this.

……

……[1]

Q 11.25)Winter 2009. P11

Study the flowchart.

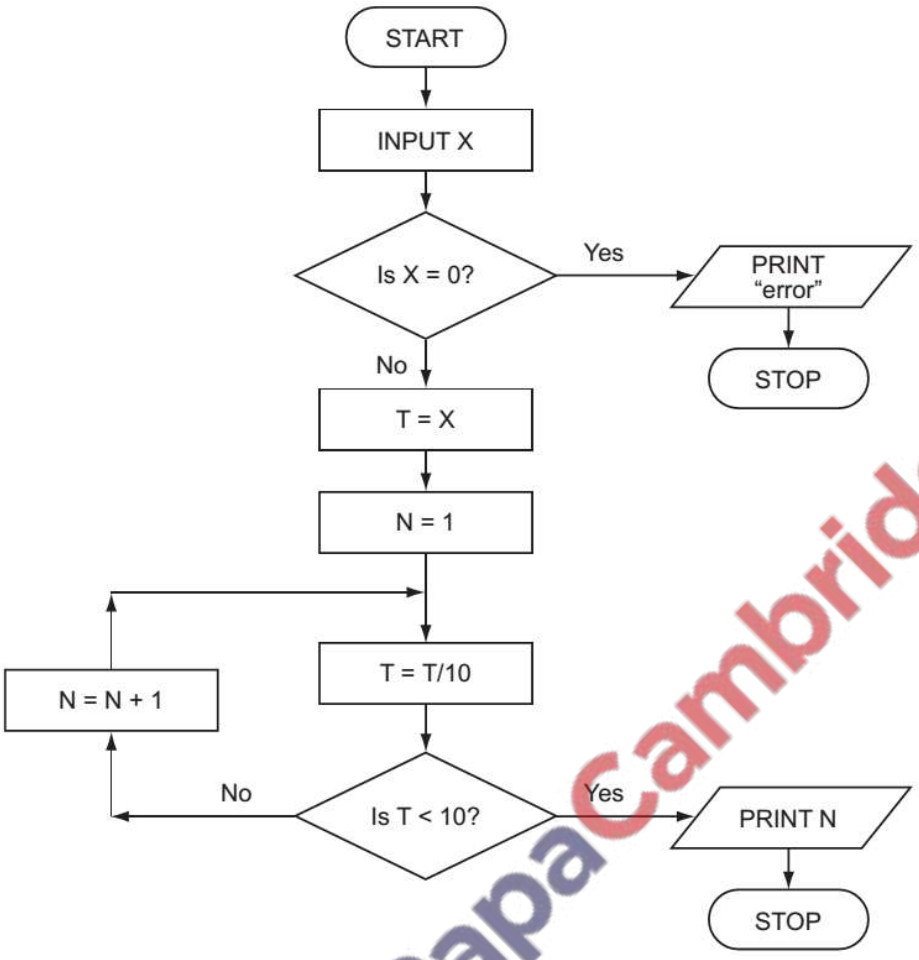


Complete the table to show what outputs you would expect for the three inputs. [3]

INPUT N	OUTPUT C
55	
2100	
1	

Q 11.26) Summer 2010 P12

Study the following flowchart very carefully:



What output would you expect if the following data was input into the flowchart? [3]

X	OUTPUT
-150	
540	
0	

Algorithm pseudo code

Q 12.1) Summer 2006 (Extract)
A formula for calculating the body mass index (BMI) is:

$$BMI = \frac{\text{weight in kilograms}}{(\text{height in metres}) \times (\text{height in metres})}$$

Using pseudo code or otherwise, write an algorithm that will input weight (kg) and height (m) of students, calculate their body mass index (BMI) and output their BMI.
Test data: 80, 2, 100, 1.9, 60, 2, 70, 1.8

First draw trace table write down column headings
Calculate BMI using trace table:

Weight	Height	BMI

Setup in pseudo code using declaration of variable
SECTION SHOWS YOU HOW THIS WOULD WORK'

Now Input Weight and height

Weight	Height	BMI
80	2	
100	1.9	
60	2	
70	1.8	

Input in pseudo code using test data

Now calculate the BMI using given formula

Weight	Height	BMI
80	2	20
100	1.9	28
60	2	15
70	1.8	22

Process in pseudo code using given formula

Now write down the above steps in pseudo code:

```
DECLARE Weight, Height, BMI: Real
INPUT Weight, Height
BMI ← Weight/(Height*Height)
OUTPUT BMI
```

Q12.2)Winter 2007 (Extract)
Fuel economy for a car is found using the formula:

$$\text{Fuel Economy} = \frac{\text{Distance Travelled (km)}}{\text{Fuel Used (litres)}}$$

Using pseudo code or otherwise, write an algorithm that will input Distance Travelled (km) and Fuel Used (litres) of cars, calculate their fuel economy and output their fuel economy.
Test data: 80, 10, 100, 5, 60, 2, 70, 5

First draw trace table write down column headings

Distance	Fuel	Fuel Economy

Setup in pseudo code using declaration of variable

Now Input Distance and Fuel

Distance	Fuel	Fuel Economy

Input in pseudo code using test data

Now calculate the Fuel Economy using given formula

Distance	Fuel	Fuel Economy

Process in pseudo code using given formula

Now write down the above steps in pseudo code:

Q12.3) Write an algorithm, using pseudo code or flowchart only, which:

- inputs real numbers
- convert them into integer (whole) numbers

(You may use INT(X) in your answer e.g. Y = INT(3.8) gives the value Y = 3)

Test data: 80.9, 10.1, 100.8, 5.6

First draw trace table write down column headings

Number X	Integer Y	Output

Setup in pseudo code using declaration of variable

Now Input Number

Number X	Integer Y	Output

Input in pseudo code using test data

Now convert the real number into whole number using INT()

Number X	Integer Y	Output

Now write down the above steps in pseudo code:

Q12.4) Write an algorithm, using pseudo code or a program flowchart only, which:

• inputs the population and land area,

• calculates the population density (i.e. population/land area),

Q12.7) This code is supposed to find out if a positive integer entered by a user is exactly divisible by the number 3.
Note: line numbers have been included and are not part of the code.

```
1       INPUT n
2       WHILE n ≥ 0
3               n ← n – 3
4       ENDWHILE
5       IF n = 0 THEN
6               OUTPUT 'is divisible by 3'
7       ELSE
8               OUTPUT 'is not divisible by 3'
9       ENDIF
```

The programmer realizes there is an error because a user input of 6 incorrectly outputs ‘is not divisible by 3'.
(a) In **Table** place a tick next to the type of error that the programmer has found. **[1]**

Type of error	Tick
Logical	
Runtime	
Syntax	

(b) State the line number of the code containing the mistake that causes this error to occur.

.....**[1]**

(c)What change needs to be made to the line of code you have identified in your answer to(b) so that the program will work correctly?

..... **[1]**

(d)What type of error could occur if the user enters the value eight?

.....**[1]**

12.7

a Logical

b2

c Any correct answer, examples include:
If the answer given for 9 (b) is 4 then
WHILE n > 0
WHILE n ≥ 1
WHILE n ≥ 3
If the answer given for 9 (a) (ii) is 7 then
IF n = -3 THEN

d Runtime error // Type error

Q12.8) The following pseudo code calculates the second hand price of different models of car.

The condition is an integer with a value between 1 and 4 where 1 is excellent and 4 is very bad.

```
INPUT Model, Condition, Age
cost ← 0
IF model = 'Daley' THEN
        cost ← 6000
ELSE IF model = 'Minty' THEN
        cost ← 4000
ELSE
        cost ← 2000
ENDIF

CASE condition OF
1: cost ← cost – 100
2: cost ← cost – 300
3: cost ← cost – 500
4: cost ← cost – 1000
ENDCASE
cost ← cost / age
PRINT cost
```

a) Tick the most appropriate data type of the variable cost. [1]

Data Type	Tick one box
Boolean	
Character	
Real	
String	

b) Complete the trace table below showing the changes in the variable cost when the following values are input: **[4]**

"Tidy", 4, 2

Cost

12.8

a) Real

b) 1 mark for every correct row that appears in the correct sequence:

cost
0
2000
1000
500

Q12.9) Write an algorithm, using pseudo code or flowchart only, which:

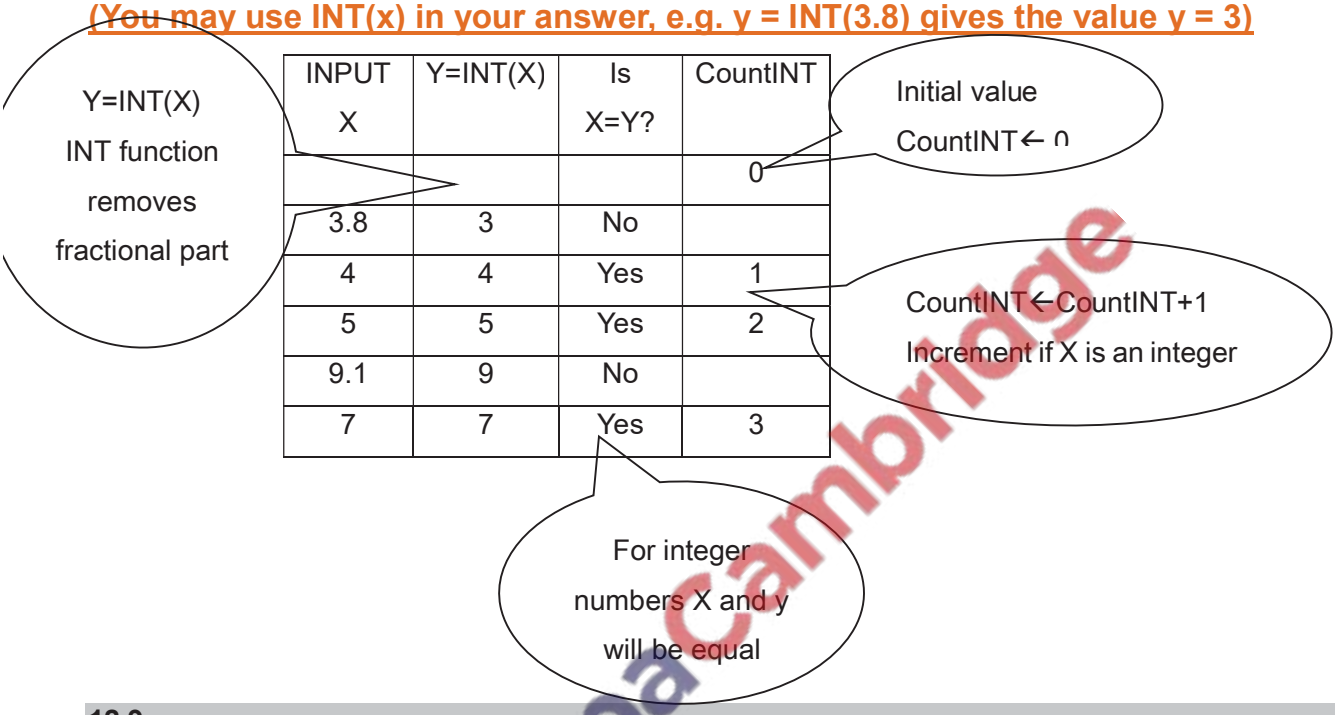
• inputs 1000 numbers

• outputs how many of these numbers were whole numbers (integers)

(You may use INT(x) in your answer, e.g. y = INT(3.8) gives the value y = 3)

..... [4]

(You may use INT(x) in your answer, e.g. y = INT(3.8) gives the value y = 3)



INPUT X	Y=INT(X)	Is X=Y?	CountINT
			0
3.8	3	No	
4	4	Yes	1
5	5	Yes	2
9.1	9	No	
7	7	Yes	3

12.9

```
DECLARE Count, CountINT : Integer
DECLARE X, Y: Real
CountINT← 0
FOR Count ← 1 TO 1000
        PRINT " Enter a number "
        INPUT X
        Y ← INT(X)
        IF X = Y THEN CountINT←CountINT + 1
NEXT Count
PRINT " Number of integers = " , CountINT
```

Q12.10)

Q12.18) A programmer uses an Integrated Development Environment (IDE) for all program development. Describe what is meant by an IDE.

.....

.....

..... [2]

12.18

IDE is a (Single) software program
Features for:
program editor/writing/editing
translation // interpreter/compiler
testing program code // observe outputs 2 points to score

Q 12.23) An algorithm to reset the contents of the array Coins after each sale is shown below. The re are 10 different coins. This algorithm contains a logic error.

```
i = 1
REPEAT
    Coins(i) = 0
    i = i + 1
UNTIL i = 10
```

(i) State what is meant by a logic error.

.....

..... **[1]**

(ii) Explain why the algorithm above contains a logic error.

.....

.....

.....

..... **[2]**

(i) •The program is written to do something other than what the programmer intended

(ii) •It will only reset the first 9 elements / will not reset the 10th element
•After setting Coins(9) = 0, i will become10...
•... and the loop will stop
•It should be UNTIL i > 10 / or other working correction

Quick Revision Questions of flowchart and pseudo code

Q9.4) Identify **three** different loop structures that you can use when writing pseudo code.

1.....

2 .....

3..... [3]

Summer 2015 P21 23

Examiner Report Question 6

Most candidates could identify at least one loop structure. A common wrong answer was to incorrectly identify IF as part of a loop structure.

Q .12 Describe the purpose of each statement in this algorithm.

```
Count ← 0
WHILE Count<10 DO
        PRINT Count
        Count ←Count + 1
ENDWHILE
```

.....
.....
.....
.....
.....[2]

9.13 Describe the purpose of each statement in this algorithm.

```
Total ← 0
INPUT Number
REAPEAT
        Total ←Total + Number
        INPUT Number
UNTIL Number < 0
PRINT Total
```

.....
.....
.....
.....
.....
.....
.....[2]

Q 9.14a) Draw a flowchart to input 20 numbers and find the average of positive numbers

Q 9.14b) Explain how do you change your flowchart to work for 30 numbers that are between 0 an d 100.

.....

.....

.....

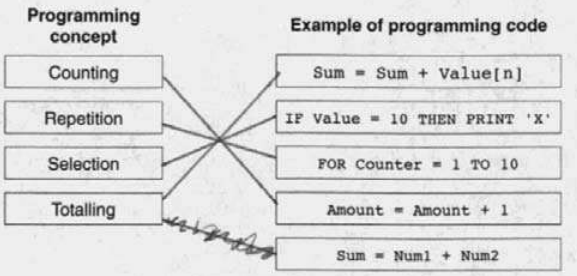
.....[3]

Candidate Example response

Example candidate response – high

4 Four programming concepts and four examples of programming code are shown below.

Draw a line to link each programming concept to the correct example of programming code.



[4]

Examiner comment – high

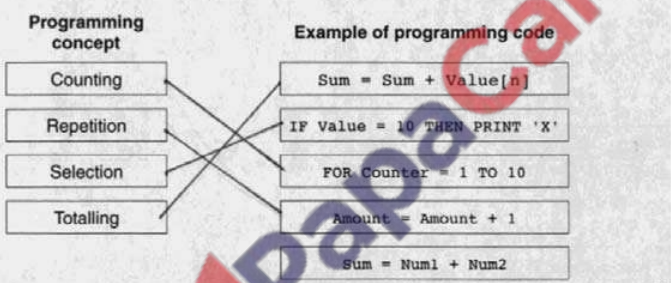
Most of the high-awarding candidates gained full marks.

Total mark awarded = 4 out of 4

Example candidate response – middle

4 Four programming concepts and four examples of programming code are shown below.

Draw a line to link each programming concept to the correct example of programming code.



[4]

Examiner comment – middle

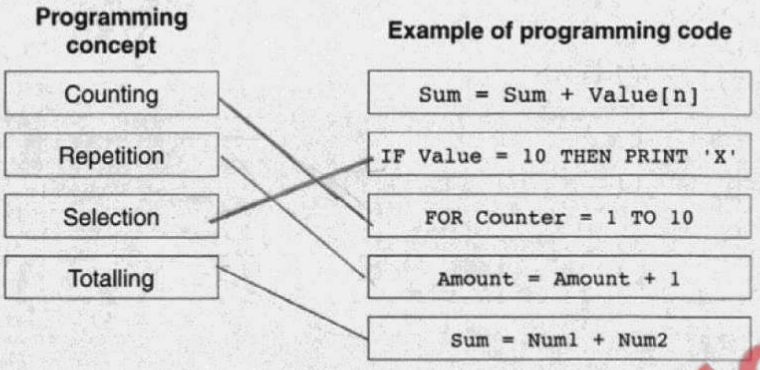
Most of the middle-awarding candidates could identify 'selection' and one other programming concept.

Total mark awarded = 2 out of 4

Example candidate response – low

4 Four programming concepts and four examples of programming code are shown below.

Draw a line to link each programming concept to the correct example of programming code.



[4]

Examiner comment – low

Most of the low-awarding candidates could only identify the programming concept of 'selection'.

Total mark awarded = 1 out of 4

Example candidate response – high

5 (a) Write an algorithm, using pseudocode and a FOR … TO … NEXT loop structure, to input 1000 numbers into an array.

```
number[1:1000]
for counter = 1 to 1000
    input num
    number[counter] = num
next counter
```

Examiner comment – high

A FOR … TO … NEXT loop with correct use of the loop counter for the array index, full marks.

Total mark awarded = 2 out of 2

Example candidate response – middle

5 (a) Write an algorithm, using pseudocode and a FOR … TO … NEXT loop structure, to inpu 1000 numbers into an array.

```
num=0
For count = 1 to 1000
Input num
Next
Numbers [1:1000] as integer
Numbers [x] <- N
```

[2

Examiner comment – middle

A FOR … TO … NEXT loop, there is no attempt to use the loop counter with the array.

Total mark awarded = 1 out of 2

5 (a) Write an algorithm, using pseudocode and a FOR ... TO ... NEXT loop structure, to input 1000 numbers into an array.

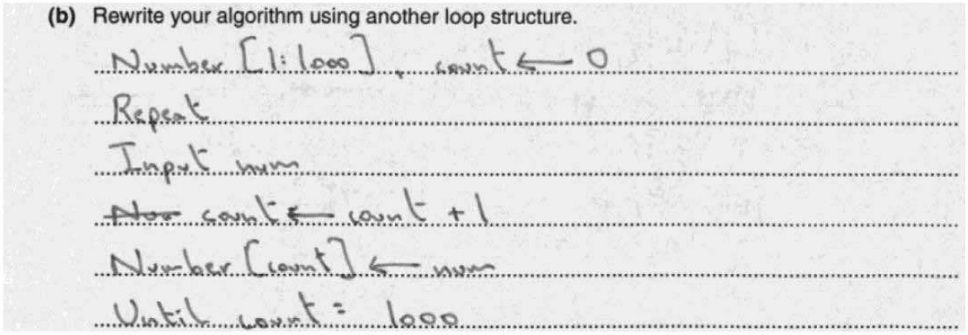
INPUT = 1000
FOR,
1000 > uPut
TO,
9999 > yiut.
Next, PRINT OutPut [2]

Examiner comment – low

An attempt at a FOR … TO … NEXT loop, there is no loop counter and no use of an array.

Total mark awarded = 0 out of 2

Example candidate response – high



(b) Rewrite your algorithm using another loop structure.

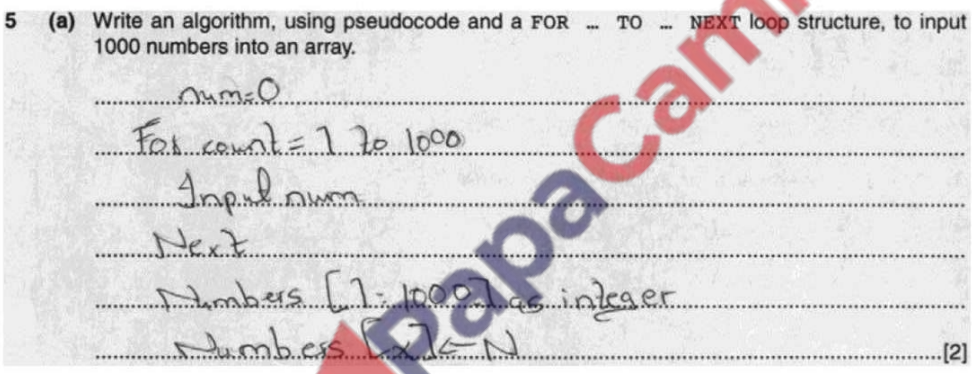
```
Number [1:1000], count ← 0
Repeat
Input num
count ← count + 1
Number [count] ← num
Until count = 1000
```

Examiner comment – high

A REPEAT … UNTIL loop, with correct initialisation, updating and testing of the loop counter, full marks. The candidate has used the correct ← symbol as required by the new syllabus. Candidates using = instead of ← were not penalised.

Total mark awarded = 4 out of 4

Example candidate response – middle



5 (a) Write an algorithm, using pseudocode and a FOR … TO … NEXT loop structure, to input 1000 numbers into an array.

```
num=0
For count = 1 To 1000
Input num
Next
Numbers [1:1000] as integer
Numbers [x] ← N
```

[2]

Examiner comment – middle

A WHILE … DO … ENDWHILE loop, with some errors. The loop counter has not been initialised, the WHILE statement is missing a variable. The updating of the loop counter is correct and there is an ENDWHILE statement, for two marks.

Total mark awarded = 2 out of 4

Example candidate response – low

(b) Rewrite your algorithm using another loop structure.

```
Numbers = [1 to 1000]
Input number
c = c + 1
Next
If the numbers = 1000 then
print Yes
display number
```

[4]

Examiner comment – low

It is unclear what type of loop is being used here. There is one mark for updating the loop counter.

Total mark awarded = 1 out of 4

Past paper questions on basic concepts of algorithm

Specimen paper 2016 P2

2 Jatinder uses Internet banking.

This pseudo code checks her PIN.

```
c ← 0
INPUT PIN
x ← PIN
REPEAT
    x ← x/10
    c ← c + 1
UNTIL x < 1
IF c <> 5
    THEN
        PRINT “error in PIN entered”
    ELSE
        PRINT “PIN OK”
ENDIF
```

(a) What value of c and what message would be output if the following PINs were entered?

5 1 0 2 0 Value of c: .....

Message:.....

5 1 2 0 Value of c:.....

Message: ..... [2]

(b) What type of validation check is being carried out here?

..... [1]

Specimen paper 2016 P2

6 (a) Write an algorithm, using pseudo code or flowchart only, which:

• inputs three numbers

• outputs the largest of the three numbers

.....

.....

.....

.....

.....

..... [3]

(b) Write an algorithm, using pseudo code or flowchart only, which:

• inputs 1000 numbers

• outputs how many of these numbers were whole numbers (integers)

(You may use INT(x) in your answer, e.g. y = INT(3.8) gives the value y = 3)

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Describe, with examples, two sets of test data you would use to test your algorithm.

1: .....

.....

2: .....

..... [2]

Summer 2016 P22

6 Identify two different selection statements that you can use when writing pseudo code.

1 .....

.....

2 .....

.....[2]

Winter 2017 P22

2 Write an algorithm using **either** pseudo code **or** a flowchart, to:

- input a positive integer
- use this value to set up how many other numbers are to be input
- input these numbers
- calculate and output the total and the average of these numbers.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

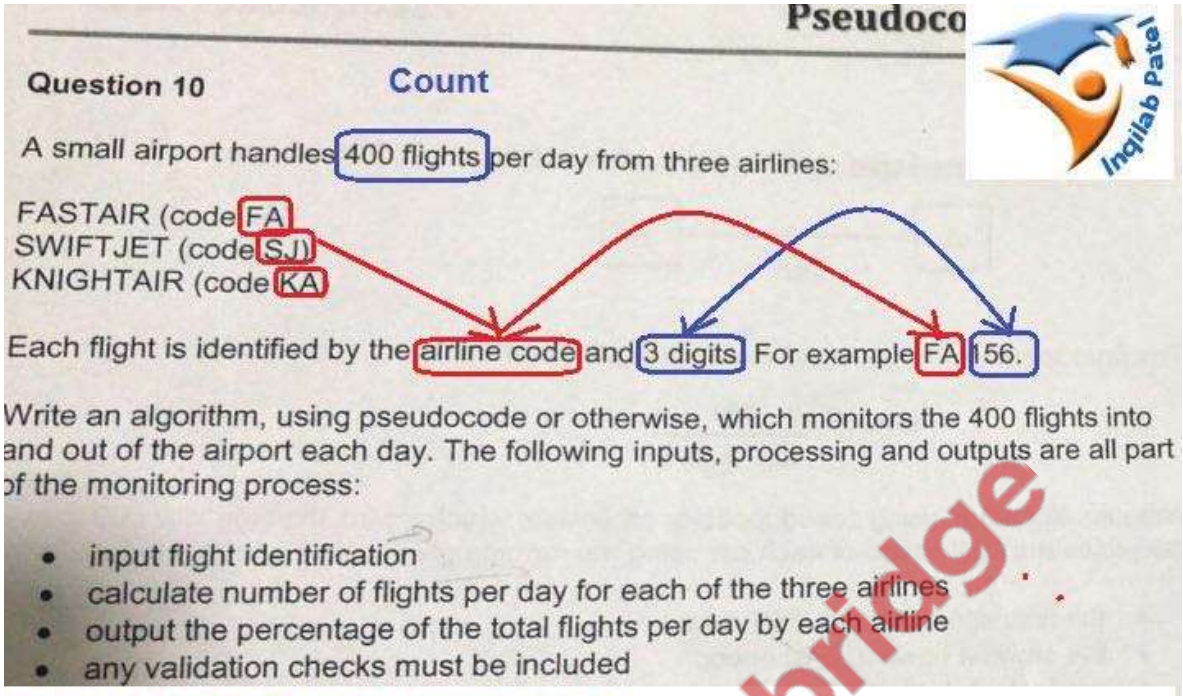
.....[6]

Winter 2017 P22

3 The following diagram shows **four** data structures and **four** descriptions. [3]

Draw a line to connect each data structure to the correct description.

Data structure	Description
Constant	A collection of related data
Array	A value that can change whilst a program is running
Table	A value that never changes whilst a program is running
Variable	A series of elements of the same data type



Pseudoco

Question 10 **Count**

A small airport handles 400 flights per day from three airlines:

FASTAIR (code FA)
SWIFTJET (code SJ)
KNIGHTAIR (code KA)

Each flight is identified by the airline code and 3 digits. For example FA 156.

Write an algorithm, using pseudocode or otherwise, which monitors the 400 flights into and out of the airport each day. The following inputs, processing and outputs are all part of the monitoring process:

- input flight identification
- calculate number of flights per day for each of the three airlines
- output the percentage of the total flights per day by each airline
- any validation checks must be included

```
DECLARE CountFA, CountSJ, CountKA: Integer
DECLARE AirlineCode, Count: Integer
CoutFA <-- 0
CoutSJ <-- 0
CoutKA <-- 0
FOR Count<-- 1 TO 400
        INPUT AirLineCode
        WHILE AirLineCode<>"FA" AND AirLineCode<>"SJ" AND  AirLineCode<>"KA" DO
                PRINT "Enter a valid air line code"
                INPUT AirLineCode
        ENDWHILE
        INPUT FlightCode
        WHILE FlightCode<100 OR FlightCode>999 DO
                PRINT" Error! Enter a valid flight code"
        ENDWHILE
        IF AirLineCode="FA" THEN CountFA <-- CountFA + 1
        IF AirLineCode="SJ" THEN CountSJ <-- CountSJ + 1
        IF AirLineCode="KA" THEN CountFA <-- Count KA + 1
NEXT Count
FAPercent <-- CountFA/400*100
SJPercent <-- CountSJ/400*100
KAPercent <-- CountKA/400*100
PRINT " Number of FastAir ", CountFA
PRINT " Number of SWIFJET ", CountSJ
PRINT " Number of KNIGHTAIR", CountKA
```

Summer 2018 P21

2 (a) Write an algorithm to input 1000 numbers. Count how many numbers are positive and how many numbers are zero. Then output the results. Use **either** pseudo code **or** a flowchart.

..... [6]

(b) Give one change you could make to your algorithm to ensure initial testing is more manageable.

..... [1]

Summer 2018 P22

2 (a) Draw a flowchart for an algorithm to input numbers. Reject any numbers that are negative and count how many numbers are positive. When the number zero is input, the process ends and the count of positive numbers is output.

(b) Explain the changes you will make to your algorithm to also count the negative numbers.

.....

.....

.....

.....[2]

Q 12.79 Winter 2018 P23

5 The algorithm allows a number to be entered. It then calculates and outputs the next number in the mathematical series.

```
Fib ←1
Prev2 ←0
Prev1 ←1

INPUT Number

IF Number = 0
        THEN Fib = 0
ENDIF
WHILE Number > 2
        Fib ←Prev2 + Prev1
        Prev2 ←Prev1
        Prev1 ←Fib
        Number ←Number-1
ENDWHILE
OUTPUT Fib
```

(a) Complete the trace table for the input data: **7** [4]

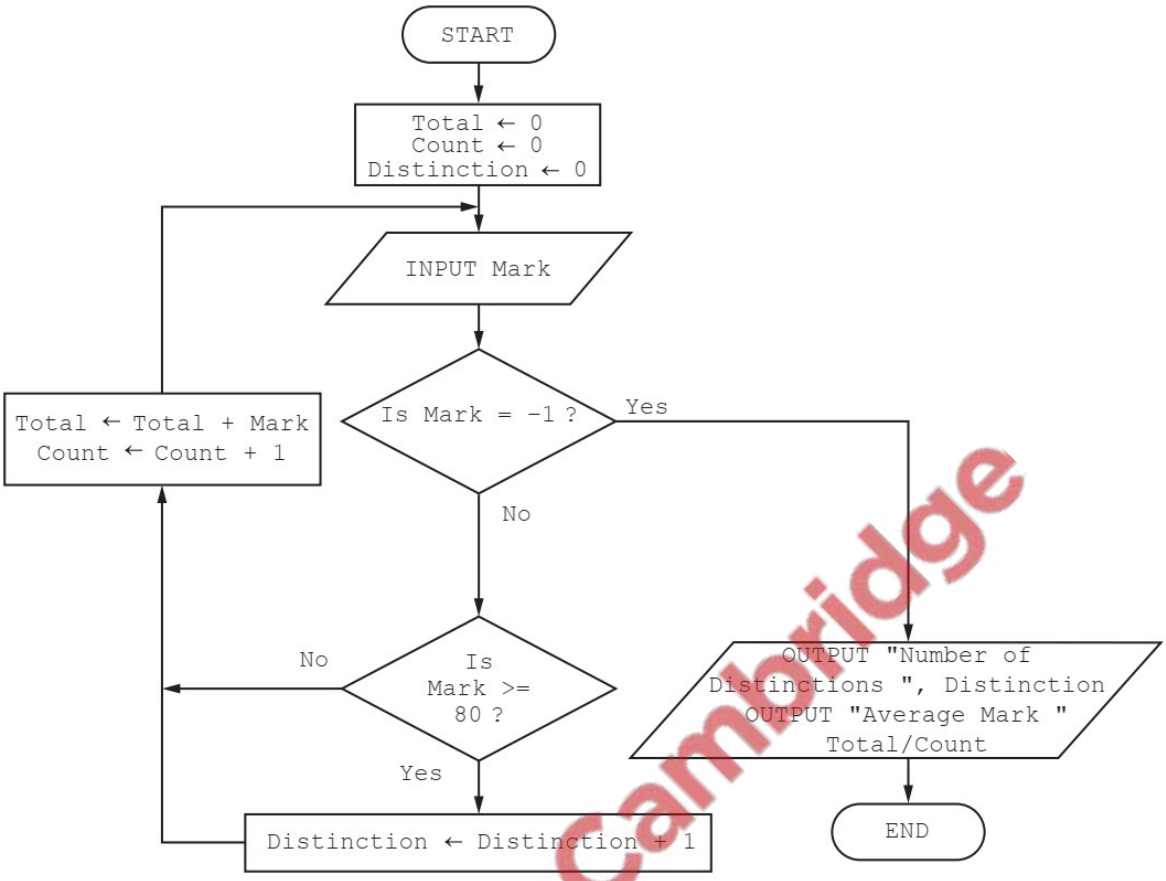
Fib	Prev2	Prev1	Number	OUTPUT

(b) Complete the trace table for the input data: **2** [2]

Fib	Prev2	Prev1	Number	OUTPUT

Q 12.83 Summer2019 P22

3 This flowchart inputs the marks gained in an examination. An input of –1 ends the routine.



Complete the trace table for the mark input data: 50, 70, 65, 30, 95, 50, 55, 85, 65, 35, –1, 45[4]

Total	Count	Distinction	Mark	**OUTPUT**

Linear Search

5 Customer names are stored in the array Customer.

An algorithm is to be designed to perform a serial search of the array for a requested customer name.

The algorithm will use the variables shown in the table.

(a) Study the table and the algorithm and fill in the gaps.

Identifier	Data Type	Description
Customer	ARRAY[100] OF STRING	Array of customer names
Index	INTEGER	Used to index the array elements
IsFound		
SearchName	STRING	The requested customer name

```
//Serial search algorithm
INPUT ……

IsFound ← FALSE
Index ← 1
REPEAT
    IF ……  =SearchName
    THEN
        IsFound ← TRUE
        OUTPUT "Found at position " Index
    ELSE
        ……..
    ENDIF
UNTIL (IsFound = TRUE) OR ……..
IF …….. THEN
    OUTPUT "Customer name was NOT FOUND"
ENDIF
```

[7]

(b) How many comparisons on average will be needed to find a requested customer from the Customer array?

.....[1]

3 (a) Customer names are stored in the array Customer.

An algorithm is to be designed to perform a serial search of the array for a requested customer na me.

The algorithm will use the variables shown in the table.

Study the table and the algorithm and fill in the gaps.

Identifier	Data Type	Description
Customer	ARRAY[2000] OF STRING	The customer names
Index	INTEGER	Index position in the customer array
IsFound		
SearchName	STRING	The requested customer name

```
//Serial search algorithm
INPUT ……
IsFound ← FALSE
Index ← 1
REPEAT
    IF Customer [ ……] = SearchName    THEN
        IsFound ← TRUE
        OUTPUT "FOUND – at position " Index " in the array"
    ELSE
        Index ← ……
    ENDIF
UNTIL (IsFound = TRUE) OR
IF ……    THEN
    OUTPUT "Customer name was NOT FOUND"
ENDIF
```

[7]

(b) Comment on the efficiency of the serial search algorithm in part (a) for retrieving a data item from an array with 2000 items.

.....

..... [2]

Errors in Pseudo code and Program

There are commonly three types of errors found in program codes:

A **syntax error** is a 'grammatical' error, in which a program statement does not follow the rules of the high-level language constructs. Due to syntax error program code can't be executed.

Logic error: an error in the logic of the solution that causes it not to behave as intended. Due to logical error a program is executed but doesn't produce required result.

Run-time error: an error that causes program execution to crash or freeze. E.g. divide-by-zero error.

Finding and correcting errors in pseudo code algorithm to Improve Efficiency

It is important to be able to identify errors and suggest corrections in a pseudo code algorithm.
If algorithm is correct but less efficient, students are asked to suggest improvements.
When task is changed, students are asked to modify pseudo code.

In loops following should points be considered:

- **Count-controlled loop** (FOR…TO…NEXT loop) should be used if number of repetition is given. For example input marks for 30 students,

```
FOR Count=1 TO 30
    INPUT marks
NEXT Count
```

- **Pre-conditioned loop** (WHILE…DO…ENDWHILE loop) should be used if loop is checked at the beginning and condition to continue the loop is given. For example to input only positive numbers, the numbers are validated at the time of input, when number is invalid, they are repeatedly input.

```
INPUT Number
WHILE Number<0 DO
    PRINT "Invalid number "
    INPUT Number
ENDWHILE
```

- **Post-conditioned loop** (REPEAT…UNTIL loop) should be used if loop is based upon a condition, but it has to be repeated at least once, and then condition to stop loop is checked. For example to input numbers, and calculate total until a rogue value like 0 is typed.

```
INPUT Number
REPEAT
    Total ←Total + Number
        INPUT Number
UNTIL Number=0
```

Common Errors in pseudo code:

There are 8 types of errors in pseudo code:

Error 1: Faulty initial or final value of loop counter

IF Count is initialized with 0 then Count< 'number of iteration' should be used in WHILE loop.
IF Count is initialized with 1 then Count<= 'number of iteration' should be used in WHILE loop.

In REPEAT… UNTIL loop the opposite of WHILE condition will be used

Example: A computer program is required which inputs 10 numbers, finally outputs the answer (the product).

WHILE Loop	
Count initialized with 0	Count initialized with 1
1 Count ← 0 2 Product← 1 3 **WHILE Count< 10 DO** 4 INPUT Num 5 Product ←Product * Num 6 Count ←Count + 1 7 **ENDWHILE** 8 PRINT Product	1 Count ← 1 2 Product← 1 3 **WHILE Count<= 10 DO** 4 INPUT Num 5 Product ←Product * Num 6 Count ←Count + 1 7 **ENDWHILE** 8 PRINT Product
REPEAT Loop	
Count initialized with 0	Count initialized with 1
1 Count ← 0 2 Product← 1 3 **REPEAT** 4 INPUT Num 5 Product ←Product * Num 6 Count ←Count + 1 7 **UNTIL Count>=10** 8 PRINT Product	1 Count ← 1 2 Product← 1 3 **REPEAT** 4 INPUT Num 5 Product ←Product * Num 6 Count ←Count + 1 7 **UNTIL Count>0** 8 PRINT Product

Example of faulty initial or final value of loop counter

To input 20 numbers and find the highest value	
1 h = 0 2 c = 0 3 REPEAT 4 READ x 5 IF x > h THEN h=x 6 c = c + 1 7 UNTIL c < 20 8 PRINT h	1 h = 0 2 c = 0 3 WHILE C<=20 4 READ x 5 IF x > h THEN h=x 6 c = c + 1 7ENDWHILE 8 PRINT h

UNTIL C >= 20

WHILE C < 20

Error 2: Missing or Faulty initialization of a variable:

A variable must be initialized if it used in calculation without INPUT.

Total is initialized with 0, Product with 1, Highest with lowest possible value and Lowest with highest possible value.

Correct Initialisation:

```
10 Count←0
20 Total ← 0
30 Product← 1
40 Highest ← 0
50 Lowest ←100
```

Example with incorrect or missing initialisation:

```
10 Count ← 0
20 REPEAT
30      INPUT Num
40      Sum ←Sum + Num
50    Count=Count + 1
50 UNTIL Count>=10
60 PRINT Sum
```

Missing Sum ←0

Error 3: Increment in loop Counter in FOR…TO…NEXT loop.

FOR…TO…NEXT loop doesn't need increment in loop counter.

Example:

```
10Sum← 0
20 FOR Count ← 1 TO 500
30      INPUT Num
40      Sum ←Sum + Num
50      Count ←Count + 1
60 NEXT Count
70 PRINT Sum
```

Remove Count ← Count + 1

Example:

```
10 Highest ← 0
20 FOR Count ← 1 TO 5
30      INPUT Num
40      IF    Num>Highest    THEN    Highest ←Num
50      Count ←Count + 1
60 NEXT Count
70 PRINT Highest
```

Error 4: Missing increment in loop Counter in REPEAT…UNTIL or WHILE…DO…ENDWHILE loop.

REPEAT…UNTIL loop and WHILE..DO..ENDWHILE loop needs increment in loop counter.

```
1 c= 0
2 h=0
3 REPEAT
4       READ x
5       IF x>h THEN h=x
6  UNTIL c>=20
7 OUTPUT h
```

Insert c ← c + 1

Error 5: Misplacing statement inside or outside of loop:

If Final output like greatest value or average is required it should be after loop.
IF running output is required it should be inside loop.

```
for count = 1 to 20 do
    input number
    if number < 0 then negative = negative + 1
    if number > 0 then positive = positive + 1
    print negative, positive
next count
```

Error 6:Missing ending keywords.
REPEAT…UNTIL or
WHILE…DO…ENDWHILE
FOR…TO…NEXT
IF…THEN…ENDIF
CASE ….OF…OTHERWISE….ENDCASE

```
1.  SET X = 1
2.  REPEAT
3.      X = X + 2
4.  Print X
```

```
set Total_1 to zero
set Counter to one
while Counter < eight
    Counter = Counter + 1
    input Number
    if Number > zero then Total_1 = Total_1 + Number
output Total_1
```

Error 7: Assignment Error.
Values or vales of variable at right side should be assigned to variables and constants at left side.
Example

```
Lowest←1000
Highest ← 0
FOR Counter ← 1 TO 100
        INPUT Number
        IF Number > Highest THEN Number ← Highest
        IF Number < Lowest THEN Number ← Lowest
NEXT Counter
PRINT Highest, Lowest
```

Error 8: Operator Error.
A common error in pseudo code is an improper operator.

Example

```
Lowest←1000
Highest ← 0
FOR Counter ← 1 TO 100
        INPUT Number
        IF Number < Highest THEN Highest ← Number
        IF Number > Lowest THEN Lowest ← Number
NEXT Counter
PRINT Highest, Lowest
```

Examination Questions

Q 13.1) Winter 2014 P13

The following pseudo code algorithm should:

• input up to 20 numbers
• stop if the sum of the input numbers **exceeds** 50
• output the final sum

```
10 count = 0
20 REPEAT
30 INPUT n
40 n + sum = sum
50 IF sum = 50 THEN count = 20
60 count = count + 1
70 UNTIL count = 20
80 OUTPUT n
```

There are **five** errors in this algorithm.
Locate these errors and suggest a correction.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

Error 5 .....

Correction .....

.....[5]

Q 13.2) Summer 2005

The following algorithm contains an error.

1. SET X = 1
2. REPEAT
3. X = X + 2
4. Print X
5. UNTIL X = 10

(a) Trace the algorithm and explain what the error is.

..... [2]

Q 13.3) Winter 2006

A computer program is required which inputs 10 numbers, multiplies them together and finally outputs the answer (the product). The following algorithm has been written to do this.

```
1       count  = 0
2       product  = 0
3       while  count<= 10 do
4               input number
5               product  = product  * number
6               count  = count  + 1
7               print product
8       endwhile
```

(a) There are three errors in the algorithm. Locate and describe these errors.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

A while do loop has been used in the algorithm. State another type of loop that could have been used.

.....

.....

Q 13.4) Winter 2010

The following algorithm inputs 20 numbers and outputs how many numbers were positive (> 0) and how many numbers were negative (< 0).

```
1       negative = 1
2       positive  = 1
3       for count  = 1 to 20 do
4               input number
5               if number < 0 then negative = negative + 1
6               if number > 0 then positive  = positive  + 1
7               count  = count  + 1
8               print negative, positive
9       next count
```

There are three different errors in this algorithm.

Locate each error and give the reason why you think it is an error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

..... [6]

Q 13.5) Summer 2011

Read the following section of code that inputs twenty (20) numbers and then outputs the largest number input.

```
1 h = 0
2 c = 0
3 REPEAT
4       READ x
5       IF x > h THEN x = h
6       c = c + 1
7       PRINT h
8 UNTIL c < 20
```

There are THREE errors in this code.

Locate these errors and suggest a corrected piece of code.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....[6]

Q 13.6) Winter 2013

A piece of pseudo code was written to input 1000 positive numbers and then output the highest and lowest numbers.

```
10 highest = 0
20 lowest = 0
30 for count = 1 to 100
40      input number
50      if number > highest then number = highest
60      if number < lowest then number = lowest
70      count = count + 1
80 next count
90 print highest, lowest
```

There are errors in the code.

Locate these errors and suggest a correction.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

Q 13.7) Winter 2014 P12

The following section of a pseudo code algorithm should:

- input 500 numbers
- generate a ratio called **k**
- output each value of **k**
- output how many numbers were larger than 10

```
10 total = 1
20 FOR x = 1 TO 500
30      IF number < 10 THEN total = total + 1
40      k = x / number
50      x = x + 1
60      OUTPUT k
70 NEXT x
80 OUTPUT x
```

(a) There are **five** errors in the above code.
Locate these errors and suggest a correction.

Error 1 .....
Correction .....
.....
Error 2 .....
Correction .....
.....
Error 3 .....
Correction .....
.....
Error 4 .....
Correction .....
.....
Error 5 .....
Correction .....
.....

(b) The corrected algorithm was converted to a computer program and run. However, after several numbers were input, the program stopped and an error message was generated, showing that there was a further error at line 40 (k = x / number).
State what could cause this error to occur.

.....
.....
Suggest a change to line 40 to overcome this problem.....
.....[2]

Candidate Example Response

Example candidate response – high

2 Read this section of program code that should input 30 positive numbers and then output the largest number input.

```
1  Large = 9999
2  Counter = 0
3  WHILE Counter > 30
4  DO
5    INPUT Num
6    IF Num > Large THEN Large = Num
7    Counter = Counter + 1
8  ENDWHILE
9  PRINT Large
```

There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

1 Line 1: The variable 'Large' should be initialised with the lowest non possible value. for eg. it's value should be set to 0.

2 In line 3, the condition set would result in loop not working and it should be 'Counter < 30'

3 In line 6, ~~if the expression~~ the condition "Num < Large" would not give correct value. It should be changed to Num > Large

4 In line 7, counter should be increased by 1. It should be counter = counter + 1. [4]

Examiner comment – high

The candidate has located all the errors correctly using the line numbers. For each error there is a correction given that would work.

Total mark awarded = 4 out of 4

Example candidate response – middle

2 Read this section of program code that should input 30 positive numbers and then output the largest number input.

```
1   Large = 9999
2   Counter = 0
3   WHILE Counter > 30
4   DO
5     INPUT Num
6     IF Num < Large THEN Large = Num
7     Counter = Counter - 1
8   ENDWHILE
9   PRINT Large
```

There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

1 line (7), it should be counter = counter + 1.

2 line (3). While counter <= 30

3 line (6). If Num > large then large = Num

4 line (1). large value is set wrong.

[4]

Examiner comment – middle

The candidate has located all the errors correctly using the line numbers. For three of the errors there is a correction given, the fourth error has no correction. The corrections for errors 1 and 3 work, the correction for error 2 will give 31 iterations not the 30 required.

Total mark awarded = 2 out of 4

Example candidate response – low

2 Read this section of program code that should input 30 positive numbers and then output the largest number input.

```
1   Large = 9999
2   Counter = 0
3   WHILE Counter > 30
4   DO
5     INPUT Num
6     IF Num < Large THEN Large = Num
7     Counter = Counter - 1
8   ENDWHILE
9   PRINT Large
```

There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

1 ~~'Num <~~ 'Large = 9999' should be "large = 1000"

2 'Counter = counter - 1' should be "counter = counter + 1"

3 "PRINT large" will come before "ENDWHILE".

4 'IF Num > large THEN large = Num' should be 'IF Num < large THEN large = Num' [4]

Examiner comment – low

The candidate has located two errors correctly by quoting the code. For each error there is a correction given, for error one the correction is wrong, for error two the correction would work. Error three is incorrect. Error four has been misidentified with the error given as the correction. Only error two has been identified and corrected.

Total mark awarded = 1 out of 4

Summer 2019 P22

2 (a) An algorithm has been written in pseudocode to input 100 numbers, select and print the largest number and smallest number.

```
Count ←1
INPUT Number
High ←Number
Low ←Count
REPEAT
    INPUT Number
    IF Number > High
    THEN
        High ←Number
    ENDIF
    IF Number > Low
    THEN
        Low ←Number
    ENDIF
    Count ←Count + 1
UNTIL Count = 99
PRINT "Largest Number is ", Number
PRINT "Smallest Number is ", Low
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1.....

Correction .....

.....

Error 2.....

Correction .....

.....

Error 3.....

Correction .....

.....

Error 4.....

Correction .....

.....[4]

(b) Show how you would change the corrected algorithm to total the numbers and print the total. Use a variable Total.

.....

.....

.....

.....

.....

.....

.....

..... [4]

Example Candidate Response – high	Examiner Comments

Section B

2 (a) An algorithm has been written in pseudocode to input 100 numbers, select and print the largest number and smallest number.

```
Count ← 1
INPUT Number
High ← Number
Low ← Count
REPEAT
  INPUT Number
  IF Number > High
    THEN
      High ← Number
  ENDIF
  IF Number > Low
    THEN
      Low ← Number
  ENDIF
  Count ← Count + 1
UNTIL Count = 99
PRINT "Largest Number is ", Number
PRINT "Smallest Number is ", Low
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1 ~~Count ← 1~~ Count ← 1

Correction Should be Count ← 0 C to have 100 ~~digits~~ inputs). 1

Error 2 Low ← Count 2

Correction Should be Low ← Number

Error 3 If Number > Low 3

Correction Should be If Number < Low

Error 4 Print "Largest Number is", Number 4

Correction Should be Print "Largest Number is", High

[4]

1 The candidate correctly identifies an error in loop count and corrects it so that 100 numbers are checked.

2 Correctly identifies error in setting up low value and corrects this to match the way the high value is set up.

3 Error in checking for low value, greater than, then correctly identifies and corrects this by changing to less than.

4 Identifies printing of the incorrect variable and corrects by replacing it with the correct variable to match the message output.

Mark for (a) = 4 out of 4

Example Candidate Response – high, continued	Examiner Comments

(b) Show how you would change the corrected algorithm to total the numbers and print the total. Use a variable Total.
Total ← 0 (5)
Count ← 0
INPUT Number
High ← Number
Low ← Number
REPEAT
INPUT Number
IF Number > High THEN
High ← Number
ENDIF
IF Number < low THEN
Low ← Number
ENDIF
Total = Total + Num (6)
Count = Count + 1
~~UNTIL Count = 99~~
UNTIL Count > 99
Print Total (7)
Print ~~High, Low~~ "largest number is", high
Print "smallest number is", low
(8)
[4]
	(5) Total correctly set to zero.
	(6) Num is added to Total instead of Number. No mark is awarded.
	(7) Total output correctly. The candidate is given a mark even although no message is output.
	(8) All changes are positioned correctly in the algorithm.
	Mark for (b) = 3 out of 4
	Total mark awarded = 7 out of 8

How the candidate could have improved their answer

- The correct variable Number should have been added to the Total instead of Num.
- A message should have been output with the variable Total, for example "Total is".

Example Candidate Response – middle	Examiner Comments

Section B

2 (a) An algorithm has been written in pseudocode to input 100 numbers, select and print the largest number and smallest number.

```
Count ← 1
INPUT Number
High ← Number
Low ← Count
REPEAT
  INPUT Number
  IF Number > High
    THEN
      High ← Number
  ENDIF
  IF Number > Low
    THEN
      Low ← Number
  ENDIF
  Count ← Count + 1
UNTIL Count = 99
PRINT "Largest Number is ", Number
PRINT "Smallest Number is ", Low
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1 INPUT Number (1)

Correction This should be omitted as it is useless and ; because it is written before REPEAT condition .

Error 2 High ← Number .

Correction Highest value should be assigned to "0" ; the lowest number . High ← 0 (2)

Error 3 Low ← count (3)

Correction Lowest number should be assigned to 100 a larger number (not stated in question) ; can be 100 .

Error 4 If Number > Low .

Correction Number should be less than Low (4) (Number < Low) for Low to be Number .

[4]

(b) Show how you would change the corrected algorithm to total the numbers and print the total. Use a variable `Total`.

~~count ← 1~~ I would assign the variable Total ~~High ← 0~~ to "0" (5) in the beginning, before ~~loop~~ the loop . After all If - - then . . . else ~~REPEAT~~ conditions I would find total as follows : Total (6) ← Total + Number and this should be written before count ← count + 1 . After the loop structure (UNTIL count > 99) I will print total as follows : PRINT "Total is (7), Total [4]

(8)

1 No error in this algorithm.

2 No error; the algorithm works when the first number to be input is used.

3 The candidate identifies the error correctly, but correction may not work, therefore, no mark is awarded.

4 Identifies error correctly, and suggests a suitable correction.

Mark for (a) = 1 out of 4

5 Total is assigned to zero.

6 Variable Total is updated correctly.

7 Correct output is added.

8 Positioning of extra statements is clearly explained.

Mark for (b) = 4 out of 4

Total mark awarded = 5 out of 8

How the candidate could have improved their answer

The candidate needed to consider the algorithm as given on the question paper and should have provided corrections for that algorithm rather than trying to rewrite the algorithm to work in a different way.

Example Candidate Response – low	Examiner Comments

Section B

2 (a) An algorithm has been written in pseudocode to input 100 numbers, select and print the largest number and smallest number.

```
Count ← 1
INPUT Number
High ← Number
Low ← Count
REPEAT
  INPUT Number
  IF Number > High
    THEN
      High ← Number
  ENDIF
  IF Number > Low
    THEN
      Low ← Number
  ENDIF
  Count ← Count + 1
UNTIL Count = 99
PRINT "Largest Number is ", Number
PRINT "Smallest Number is ", Low
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1 ..If number >= High.. 1

Correction ..If number>=High..

Error 2 ..If Number >low.. 2

Correction ..If Number <= low..

Error 3 ..UNTIL Count > 99.. 3

Correction ..UNTIL Count <= 100..

Error 4 ..Print Statments..

Correction ..It must be before loop.. 4

[4]

1 No correction is required for this statement.

2 Error is correctly identified, and the correction works even although it is not efficient, as an equal value would be replaced by the same value.

3 Correction is incorrect as the loop would only iterate once rather than 100 times.

4 No error.

Mark for (a) = 1 out of 4

Example Candidate Response – low, continued	Examiner Comments

(b) Show how you would change the corrected algorithm to total the numbers and print the total. Use a variable `Total`.
Total ← 1 (5)
InputNumber
If number >= total then
Total ← number
Print Total (6)
Endif
Total ← Total + 1 (7)
Until Total ← Total numbers. [4]
(8)
	(5) Total should be set to zero.
	(6) Entry 3 mark is awarded without the message.
	(7) Number should be added to Total not 1.
	(8) Position of Print Statement is incorrect; no mark awarded.
	Mark for (b) = 1 out of 4
	Total mark awarded = 2 out of 8

How the candidate could have improved their answer

- **(a)** The candidate needed to carefully consider how the algorithm used the selection statements as errors were incorrectly identified (error 1) or the correction could have been improved (errors 2 and 3). The candidate needed to have realised that the print statements were in the correct position after the numbers had been checked and looked for another error in the print statements.
- **(b)** The candidate needed to use totalling rather than counting for the solution. The Total needed to be printed at the end of the algorithm. The candidate needed to update the original algorithm instead of writing a new algorithm.

Common mistakes candidates made in this question

- **(a)** Candidates tried to rewrite rather than correct the algorithm.
- **(b)** Candidates wrote a new algorithm rather than using the one provided. Candidates did not output a message with the value of `Total`.

Questions from Past Papers

13.8 Summer 2015 P21& 23

2 Read this section of program code that should input 10 positive numbers and then output the smallest number input.

```
1 Small = 0
2 Counter = 0
3 REPEAT
4     INPUT Num
5     IF Num< Small THEN Num = Small
6     Counter = Counter + 1
7     PRINT Small
8 UNTIL Counter < 10
```

There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

..... [4]

Examiner Report Question 2

Most candidates located at least one error and suggested a suitable piece of corrected code. The error on line 8 was often identified, with better candidates providing a working correction.

13.9 Summer 2015 P22

2 Read this section of program code that should input 30 positive numbers and then output the largest number input.

```
1 Large = 9999
2 Counter = 0
3 WHILE Counter > 30
4 DO
5       INPUT Num
6       IF Num< Large THEN Large = Num
7       Counter = Counter - 1
8 ENDWHILE
9 PRINT Large
```

There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

..... [4]

Examiner's comments on Question 2

Most candidates located at least one error and suggested a suitable piece of corrected code. The error on line seven was the one identified and corrected by nearly all candidates. The error on line 3 was often identified, with better candidates providing a working correction.

13.10 Winter 2015 P21 & 22

2 Read this section of program code that should input 50 numbers and then output the average.

```
1 Total = 0
2 For Counter = 1 TO 50
3       INPUT Num
4       Total = Total + 1
5       Counter = Counter + 1
6       Average = Total/Counter
7 NEXT Counter
8 PRINT Average
```

There are **four** errors in this code. Locate these errors and suggest code corrections to remove each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....[4]

Examiners' Comments Question 2

Many candidates located at least one error and suggested a suitable piece of corrected code. The errors on lines 4 and 5 were frequently identified, with stronger responses providing a working correction. The question asked the candidates to identify and correct each error; a few candidates either identified the error or corrected the error, but both actions were required to gain each mark.

13.11 Summer 2016 P22

2 Read this section of program code that inputs 10 positive numbers and then outputs the total.

```
1 Total = 0
2 Counter = 0
3 REPEAT
4       INPUT Num
5       Total = Total + Num
6       PRINT Total
7       Counter = Counter + 1
8 UNTIL Counter = 10
```

This code works, but it is inefficient.

(i) Suggest **three** improvements that could be made.

1 .....

.....

2 .....

.....

3 .....

.....[3]

(ii) Rewrite the program code with your improvements.

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

13.12 Specimen paper 2016 P2

4 Read this section of program code that inputs twenty (20) numbers and then outputs the largest number input.

```
1 h = 0
2 c = 0
3 REPEAT
4       READ x
5       IF x > h THEN x = h
6       c = c + 1
7       PRINT h
8 UNTIL c < 20
```

There are three errors in this code.

Locate these errors and suggest a corrected piece of code.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....[3]

13.13 Winter 2016 P21-23

2 Read this section of program code that:

• inputs 10 numbers

• checks whether each number is within a specified range

• totals the numbers within the range and outside the range

```
1 InRange = 0
2 OutRange = 1000
3 FOR Count = 1 TO 10
4       INPUT Num
5       IF Num> 10 AND Num< 20 THEN InRange = InRange + 1
6       ELSE OutRange = OutRange - 1
7       Count = Count + 1
8 NEXT X
9 PRINT InRange, OutRange
```

(a) There are four errors in this code.

Locate these errors and suggest a correction to remove each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....[4]

(b) Decide, with reasons, whether the numbers 10 and 20 are within or outside the range. [4]

Number	Withinrange (✓)	Outside range (✓)	Reason
10			....
20			....

13.14 Winter 2016 P22

2 Read this section of program code that inputs positive numbers, discards any negative numbers and then outputs the average. An input of zero ends the process.

```
1 Total = 0
2 Counter = 100
3 REPEAT
4       REPEAT
5               INPUT Num
6       UNTIL Num< 0
7       Total = Total + 1
8       Counter = Counter + Num
9 UNTIL Num = 0
10 Average = Total / (Counter - 1)
11 Print Average
```

There are four errors in this code.

Locate these errors and suggest a correction to remove each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....[8]

13.15 March 2017 P21 (India)

2 Read this section of code that inputs the ages of people entering an event. The input sequence is ended by inputting a negative value for age. The code outputs the number of people at the event over the age of 18.

```
01    Num18 = 0
02    INPUT Age
03    WHILE Age >= 0 DO
04        IF Age >= 18 THEN
05            Num18 = Num18 + Age
06        ENDIF
07    ENDWHILE
08    PRINT Num18 – Age
```

There are four errors in this code.

Locate these errors and suggest code correction to remove each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....[4]

13.16 Summer 2017 P21

2 This section of program code asks for 50 numbers to be entered. The total and average of the numbers are calculated.

```
1 Total = 0
2 Counter = 50
3 PRINT 'When prompted, enter 50 numbers, one at a time'
4 REPEAT
5     PRINT 'Enter a number'
6     INPUT Number
7     Total + Number = Total
8     Number = Number + 1
9 UNTIL Counter = 50
10 Average = Number * Counter
11 PRINT 'The average of the numbers you entered is ', Average
```

There are **four** errors in this code.

State the line number for each error and write the correct code for that line.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

..... [4]

13.17 Summer 2017 P22

4 An algorithm has been written in pseudo code to input 100 numbers and print out the sum.

A REPEAT … UNTIL loop has been used.

```
Count ←  0
Sum ←  0
REPEAT
    INPUT Number
    Sum ← Sum + Number
    Count ←Count + 1
UNTIL Count > 100
PRINT Sum
```

(a) Find the error in the pseudo code and suggest a correction.

Error 1 .....

Correction .....

.....[2]

(b) Rewrite the correct algorithm using a more suitable loop structure.

.....

.....

.....

.....

.....

.....

..... [3]

13.18 Winter 2017 P21

2 This section of program code asks for 80 numbers between 100 and 1000 to be entered. It checks that the numbers are in the correct range, and stores them in an array. It counts how many of the numbers are larger than 500 and then outputs the result when the program is finished.

```
1 Count = 0
2 FOR Index = 1 TO 80
3       INPUT 'Enter a number between 100 and 1000', Number
4       WHILE Number = 99 AND Number = 1001
5             INPUT 'This is incorrect, please try again', Number
6       ENDWHILE
7       Num[80] = Number
8       IF Number > 500 THEN Count = Count + 1
9 UNTIL Index = 80
10 PRINT Index
11 PRINT ' numbers were larger than 500'
```

There are **four** lines of code that contain errors.

State the line number for each error and write the correct code for that line.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....[4]

13.19 March 2018 P22 (India)

2 An algorithm has been written in pseudo code to input some numbers and print out any numbers that are greater than or equal to 100. The number 999 stops the algorithm.

```
INPUT Number
WHILE NUMBERS <> 999 DO
        IF Number > 100 THEN PRINT Number ENDIF
ENDWHILE
PRINT Number
```

(a) Find the **four** errors in the pseudo code and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....[4]

(b) Show, using pseudo code, how you would change the corrected algorithm to print out any numbers between 100 and 200 inclusive.

.....

.....

.....

.....

.....

.....

.....[2]

Comments on Question 2

(a) *Most candidates correctly identified one or two errors. A few candidates showed good understanding of the pseudo code by correctly identifying the problem with the variable name and the need to add INPUT Number before ENDWHIL E. A common error was to suggest that the WHILE condition was incorrect.*

(b) *Some candidates realised that as well as introducing an upper bound, there was a change required to the value of t he lower bound of the selection test, as the number 100 would now be included.*

13.20Winter 2018 P23

4 This is a section of program code.

```
1 Total = 100.00
2 PRINT 'Enter the height of each member of your class, one at a time, when prompted'
3 FOR Count = 1 TO 30
4     PRINT 'Enter a height in metres'
5     INPUT Height
6     Total = Total + Height
7     PRINT Total / 30
8     Count = Count + 1
9 NEXT Count
```

(a) There are **three** errors in this code.

State the line numbers that contain the errors and describe how to correct each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....[3]

(b) State the purpose of this program.

.....

.....

.....[1]

13.21 March 2019 P22 (India)

2 (a) An algorithm has been written in pseudocode to input 50 numbers and total only the positive numbers.

```
Count ←1
Total ← Count
REPEAT
    INPUT Number
    IF Number <> 0
    THEN
        Total ← Total + Count
    ENDIF
    Count ← Count + 1
UNTIL Count < 50
PRINT Total
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

..... [4]

(b) Show how you would change the corrected algorithm to only total numbers greater than 0and less than 20.

.....

.....

..... [2]

13.22 Summer 2019 P22

2 (a) An algorithm has been written in pseudocode to input 100 numbers, select and print the largest number and smallest number.

```
Count ←1
INPUT Number
High ←Number
Low ←Count
REPEAT
    INPUT Number
    IF Number > High
    THEN
        High ←Number
    ENDIF
    IF Number > Low
    THEN
        Low ←Number
    ENDIF
    Count ←Count + 1
UNTIL Count = 99
PRINT "Largest Number is ", Number
PRINT "Smallest Number is ", Low
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1.....

Correction .....

.....

Error 2.....

Correction .....

.....

Error 3.....

Correction .....

.....

Error 4.....

Correction .....

.....[4]

(b) Show how you would change the corrected algorithm to total the numbers and print the total. Use a variable Total.

.....

.....

.....

.....

.....

.....

.....

..... [4]

Chapter 14

Arrays: Data Structure & Pre-Release Materials

Arrays

Arrays are data structure used to store multiple data items of same data type under one identifier name.

Arrays are considered to be fixed-length structures of elements of identical data type, accessible by consecutive index (subscript) numbers. It is good practice to explicitly state what the lower bound of the array (i.e. the index of the first element) is because this defaults to either 0 or 1 in different systems. **Generally, a lower bound of 1 will be used**.

Square brackets are used to indicate the array indices.

Each element in the array is identified using its **subscript** or **index number**. The largest and smallest index numbers are called the *upper bound* and *lower bound* of the array.

Example

StudentName[1:30]

For illustration, let's take array declaration to store marks of 10 students.

Marks[1:10]

After storing values in array

As per the above illustration, following are the important points to be considered.

- Index starts with 1.
- Array length is 10 which means it can store 10 elements.
- Each element can be accessed via its index. For example, we can fetch an element at index 6 as 19.

The terms associated with Arrays

Name: The identifier of the array is called Array Name. E.g. StudentName[]

Element: Each data item stored in arrayis called element. Array can store only single types of elements.

Size: The number elements the array can store. E.g. StudentName[1:30] can store 30 names while StudentName[30] can store 31 names as by default it is 0 to 30.

Index: The position of each element is referred as Index Number. Index of Abdullah in array example is 1.

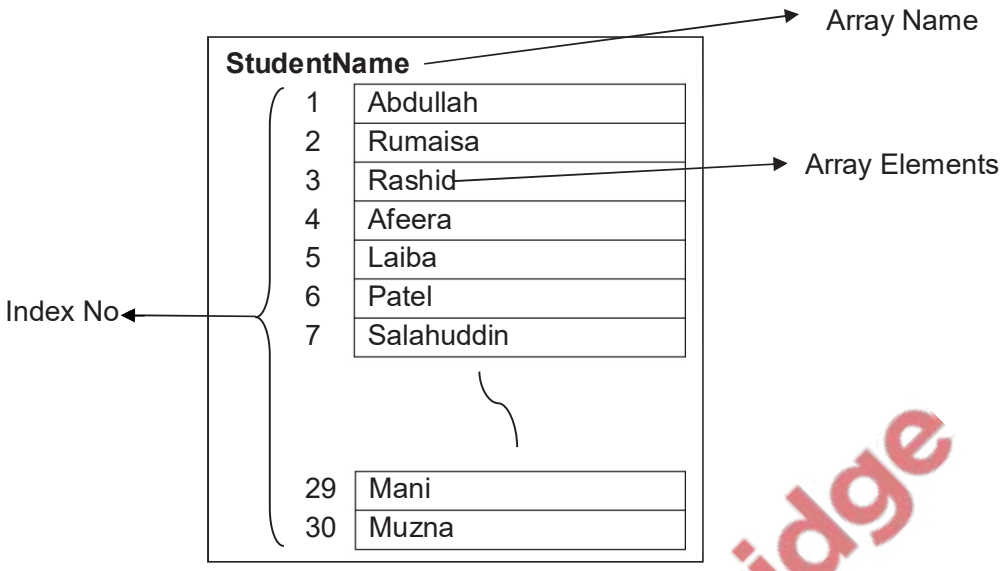
Type: Data type of all elements in a single array have same data types.

Dimension: Dimension is the organisational structure of array. It may be 1D that has single column or 2D that have multiple columns.

Example

DECLARE StudentName[1:30] : STRING

StudentName[1] ← “Abdullah”



Declaring an array

It is important declare the arrays before assigning values in it so that program can reserve that amount of space in its memory; otherwise, there may not be enough space when the program uses the data.

Declaration consists of telling the computer program:

- the identifier name of the array
- the sort of data that is going to be stored in the array, i.e. its data type
- How many items of data are going to be stored, so that it knows how much space to reserve.

Different programming languages have different statements for initialising the array but they all do the same thing. In Visual Basic, the statement is:

Dim Name(20) As String

This Dim statement declares:

- the identifier name: Name
- the upper bound: 20
- the data type: String.

The upper bound of 20 specifies that there can be a maximum of 21 data items, since Visual Basic starts with a subscript of zero. We do not have to fill the array; the upper bound of 20 indicates the maximum size.

The array that has been described in one dimension array so far is really only a list of single data items. It is possible to have an array which can be visualised as a two-dimensional table with rows and columns and a data value in each cell.

Reading data into an array

To assign data values to the elements of the array, we do this with assignment statements such as:

Name(6) = “Patel”

This places the string “Patel” at index position 6 in the array.

Similarly, the following statement places the string “Rashid” at index position 3 in the array.

Name(19) = “Mani”

Quick Revision Questions

Q 17.1) Explain the following terms regarding arrays:

Array: ……

……

Size of Array: ……..

……

Element: ……

……

Index: ……

……

Type: ……

……

Dimension: ……

……

Q 17.2) Explain with the help of examples when arrays are used in programming.

……

……

……

……

……

……

Q 17.3) Declare arrays to Explain with the help of examples when arrays are used in programming .

a) Declare arrays to store name of 30 students

……

……

b) Declare arrays to store basic pay of 50 Employees.

……

……

c) Declare arrays to input and store status of 50 employee that they are permanent or not.

……

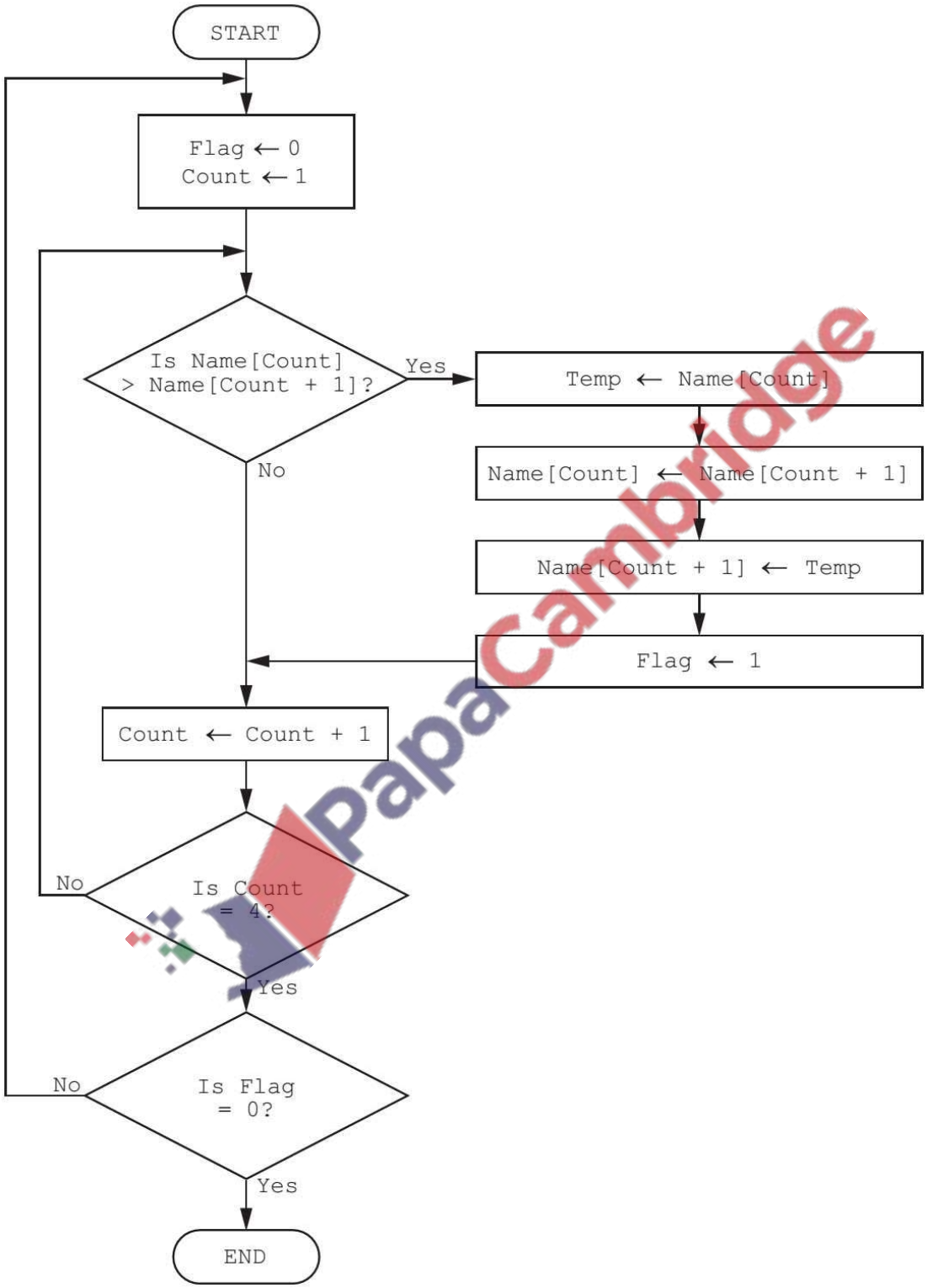
……

Q 17.4) Draw a flowchart that

- Inputs name of 10 students in a class and store them in one dimension array
- Display list of names of students

Past paper flowchart for same type of question in Winter 2017 P21 Q5

The flowchart below represents a program routine.



(a) The array used in the flowchart contains the following data:

Name[1]	Name[2]	Name[3]	Name[4]

Jamal	Amir	Eve	Tara

Complete the trace table using the data given in the array. [5]

Flag	Count	Name[1]	Name[2]	Name[3]	Name[4]	Temp
		Jamal	Amir	Eve	Tara	

(b) Describe what the algorithm represented by the flowchart is doing.

.....

.....[2]

Q 17.5Summer 2018 P21

3 The global trade item number (GTIN-8) barcode has seven digits and a check digit.
This pseudocode algorithm inputs seven digits and calculates the eighth digit, then outputs theGTIN-8.
DIV(X,Y), finds the number of divides in division for example **DIV(23,10)** is 2.
MOD(X,Y), finds the remainder in division for example **MOD(23,10)** is 3.

```
FOR Count ← 1 TO 7
   INPUT Number
   Digit(Count) ← Number
NEXT
Sum ← (Digit(1)+Digit(3)+Digit(5)+Digit(7))*3+Digit(2)+Digit(4)+Digit(6)
IF MOD(Sum,10) <> 0
   THEN Digit(8) ← DIV(Sum,10)*10 + 10 - Sum
   ELSE Digit(8) ← 0
ENDIF
OUTPUT "GTIN-8"
FOR Count ← 1 TO 8
   OUTPUT Digit(Count)
NEXT
```

(a) Complete the trace table for the input data: 5, 7, 0, 1, 2, 3, 4 [5]

Digit (1)	Digit (2)	Digit (3)	Digit (4)	Digit (5)	Digit (6)	Digit (7)	Digit (8)	Sum	OUTPUT

Complete the trace table for the input data: 4, 3, 1, 0, 2, 3, 1

Digit (1)	Digit (2)	Digit (3)	Digit (4)	Digit (5)	Digit (6)	Digit (7)	Digit (8)	Sum	OUTPUT

(b) Explain how you would change the algorithm to input eight digits (seven digits and the check-digit) and output if the check digit entered is correct or not.

.....

.....

.....

.....

.....

.....

..... [3]

Q 17.6) An algorithm to reset the contents of the array Coins after each sale is shown below. Ther e are 10 different coins. This algorithm contains a logic error.

```
i = 1
REPEAT
        Coins(i) = 0
        i = i + 1
UNTIL i = 10
```

(i) State what is meant by a logic error.

.....

..... **[1]**

(ii) Explain why the algorithm above contains a logic error.

.....

.....

.....

..... **[2]**

(i) •The program is written to do something other than what the programmer intended

(ii) •It will only reset the first 9 elements / will not resetthe 10th element
•After setting Coins(9) = 0, i will become10...
•... and the loop will stop
•It should be UNTIL i > 10 / or other working correction

Q 17.8Summer 2015 P22

5 (a) Write an algorithm, using pseudo code and a FOR … TO … NEXT loop structure, to input 1000 numbers into an array.

.....

.....

.....

.....

.....

.....[2]

(b) Rewrite your algorithm using another loop structure.

.....

.....

.....

.....

.....

.....[4]

Examiner's comments on Question 5
(a) Most candidates attempted the loop structure, better candidates also showed the skill of being able to use the loop counter as the array index. Some candidates misread the question and incorrectly provided program code rather than pseudo code.
(b) Better candidates correctly used REPEAT … UNTIL or WHILE … DO … ENDWHILE structures.

The most challenging aspect was the correct management of the loop counter.

Q 17.9Summer 2016 P22

5 A programmer writes a program to store a patient's temperature every hour for a day. State the data structure that would be most suitable to use and give the reason for your choice.
Data structure .....
Reason .....
.....[2]

5 (a) Describe the purpose of each statement in this algorithm.

```
FORI← 1 to 300
        INPUT Name[I]
NEXT I
```

.....
.....
.....
.....
.....[2]

(b) Identify, using pseudocode, another loop structure that the algorithm in **part (a)** could have used.

.....
.....[1]

(c) Write an algorithm, using pseudocode, to input a number between 0 and 100 inclusive. The algorithm should prompt for the input and output an error message if the number is outside this range.

.....
.....
.....
.....
.....
.....
.....
.....[3]

Q 17.10Winter 2017 P22

3 The following diagram shows **four** data structures and **four** descriptions. [3]

Draw a line to connect each data structure to the correct description.

Data structure	Description
Constant	A collection of related data

Array	A value that can change whilst a program is running
Table	A value that never changes whilst a program is running
Variable	A series of elements of the same data type

A restaurant table will have its data stored in its own booking record. Alessio decides to use an array of records.

Write **program code** to declare the array TableBookingsfor the 12 table records.

Programming language.....

Code .....

.....[1]

Summer 2016 P21 &P23

(ii) The swimming club has 50 members.

State the data structure that would be most suitable to use and give a reason for your choice.

Data structure .....

Reason .....

.....[2]

Practice Questions

Q 1a) Declare an array called Marks whose index values ranges from 0 to 999 and whose element type is integer

.....

.....[1]

b) Initialise the array declared in part a, all values in the array should be equals to 0.

.....

.....

.....

.....

.....[2]

Q 2) Consider the following code:

DECLARE NextChar[1:30] : CHAR

Write a code to store the letter 'A' at 1st and 'Z' at 26th location of the Array.

.....

.....

.....

.....

.....[2]

Q3a) Write a pseudo code that uses an array to store marks of 10 students of a class. Enter marks of each student. After input all the marks output the list of marks.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

b) Calculate the average marks of the class by traversing the array.

.....

.....

.....

.....

.....

.....

.....

.....[3]

c) Output the smallest and the greatest marks of the class by traversing the array.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

d) Create a second array to input and store name of students of the class. Output the list of name of students and their marks by traversing the two arrays. At the end of list print average marks of the class.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

Q 4a) Write a program to input name and telephone of your friends in two one dimension arrays. Each array can store up to 5 elements.

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

b) Write a program that inputs the name of your friend search in the array and output telephone number.

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

c) After the user selects a name, give him option to display and change the telephone number in the array. After change of telephone number, output the entire list.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

Q 5) John works in a supermarket, where he is given a task to find which item has the highest pric e and which item has the lowest price. There are 900 items in the supermarket.

a. Declare suitable arrays to store name and price of each product.
b. In put price of each product with its name.

c. Output the number of items which have a price greater than 100and number of items which have price less than 50.

d. Output the highest and the lowest price.

……

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Chapter 15

Database

2.3 Database

- define a single-table database from given data storage requirements
- choose and specify suitable data types
- choose a suitable primary key for a database table
- perform a query-by-example from given search criteria

A database is an organized and persistent (permanent) collection of data.

The collected information could be in any number of formats (electronic, printed, graphic, audio, statistical, combinations). There are physical (paper/print) and electronic databases.

A database could be as simple as an alphabetical arrangement of names in an address book or as complex as a database that provides information in a combination of formats.

Examples:

- phone book
- address book
- Census Bureau data

Database Management System (DBMS)

Database management system is a mechanism for manipulating data with high level command. It hides low level details such are as how data are obtained.

Database management system also has ability to search record by queries and to create reports and view data.

Entity

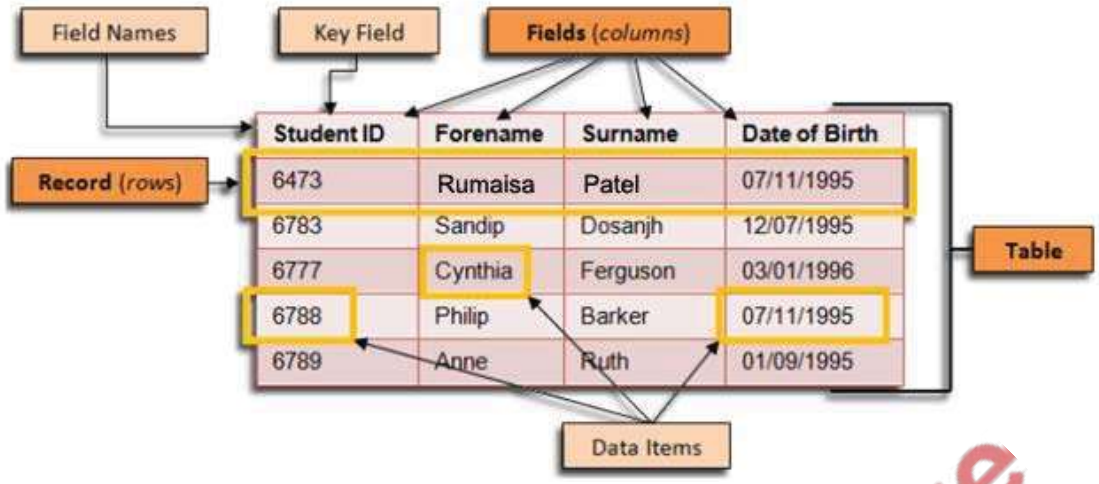
An entity is a "real world thing" about which data is held. Examples of entities include:

A customer	A product	A pupil	A supplier
A hotel room	A DVD	A flight	A holiday
A treatment	An address book	A book	A car
An order	An animal	A student	

An attribute is a feature of that entity. For example, a hotel room might have an attribute about whether it has a view or whether it is single or double. A student might have a date of birth and an address.

An entity is stored as a table in a database and an attribute becomes a field in a table.

All the data about a particular entity is stored in a single table. Each data item about the entity is a field.



Database record

Data in a database table is organised into rows **(records)** and columns **(fields)**. Each record in a relational database table corresponds to an entity. In the example table of 'Students' above there are 5 records. Each record corresponds to an individual student. Note that although there are two students called Philip Barker with the same date of birth, they have different Student IDs and are different students.

Database field

An attribute is a piece of information or a characteristic of an entity. Attributes of entities are represented in database tables by **fields** (columns). A field stores one item of data for a record. In the table above, each student is represented in the relational database by a record and the student attributes are stored in the following fields:

- Student ID
- Forename
- Surname
- Date of Birth

Fields have the following characteristics:

- Each field in a table has a unique name. Note, however, that the same field name can occur in other tables of the same relational database.
- Each field stores a single item of data - For example, a field called Date of Birth would store no more than one date of birth value.
- Each field has a particular data type – for example, text, Boolean, integer, date/time, etc.
- Each field can have its own validation rules - these ensure that data recorded in the field is of the right type and format.

Data types

Different data types are identified so that a computer can store and process the data appropriately. Data types include:

- text ()
- number (numeric) may include:
    - Auto number
    - Currency
- date/time
- Boolean (or Yes/No).

Primary Keys

Each table has a primary key. This is a field chosen so that it can uniquely identify each record. Sometimes an existing attribute can be used because it is unique but most of the time some sort of ID is created. Primary keys can be used to link to foreign keys in other tables. A foreign key is the primary key in a different table and it is not necessarily unique.

Example Question:

A picture gallery owner has decided to set up a database to keep information about the pictures he has for sale. The database table, PICTURE, will contain the following fields:

Title; Artist; Description; Catalogue Number; Size (area in square centimeters); Price; Arrived (date picture arrived at gallery); Sold (whether picture is already sold)

(a) (i) State what data type you would choose for each field.

Title: .....

Artist: .....

Description: .....

Catalogue Number: .....

Size: .....

Price: .....

Arrived: .....

Sold: .....[4]

(ii) State which field you would choose for the primary key

……[1]

Query

The prime function of a relational database is to store data in an organised way so that users can interrogate (search) and manipulate (sort) the data. The interrogation of a database is called querying the database and a question used to interrogate the data is called a query.

Query by Example (**QBE**) is a database **query** language for relational databases. It was devised by Moshé M. Zloof at IBM Research during the mid-1970s, in parallel to the development of SQL. It is the first graphical **query** language, using visual tables where the user would enter commands, **example** elements and conditions.

Databaseuser-interface in which the user fills out a form to retrieve data. The database makes the search on the basis of the example(s) provided by the user.

The query to find students in form 10B

A complex query looks for data in two or more fields and uses the logical operators OR, AND or NOT.

The following example uses a complex query to find all of the pupils in Form 10B who were born before 1995. This query uses the logical operator AND:

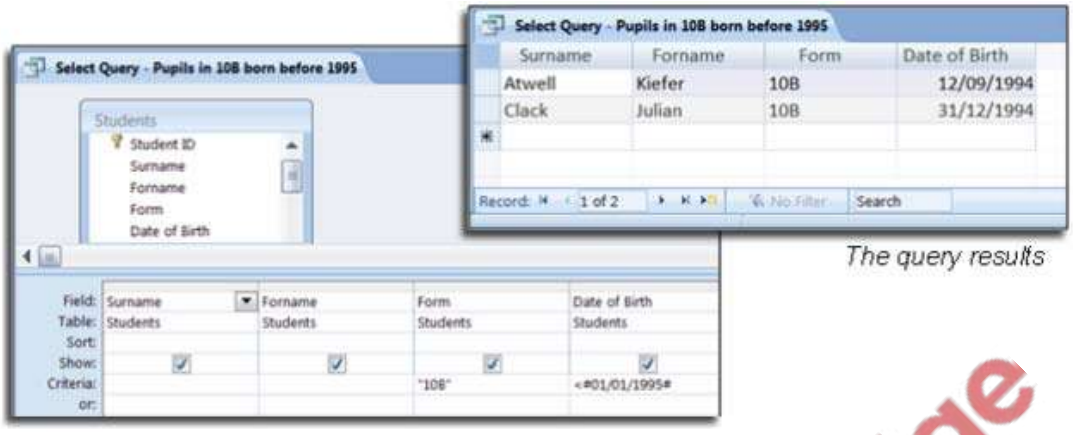
(Form = "10B") AND (Date of Birth < 01/01/1995).

Operators can be used to refine search results.

Operator	Meaning
=	Equals
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<>	Not equal to

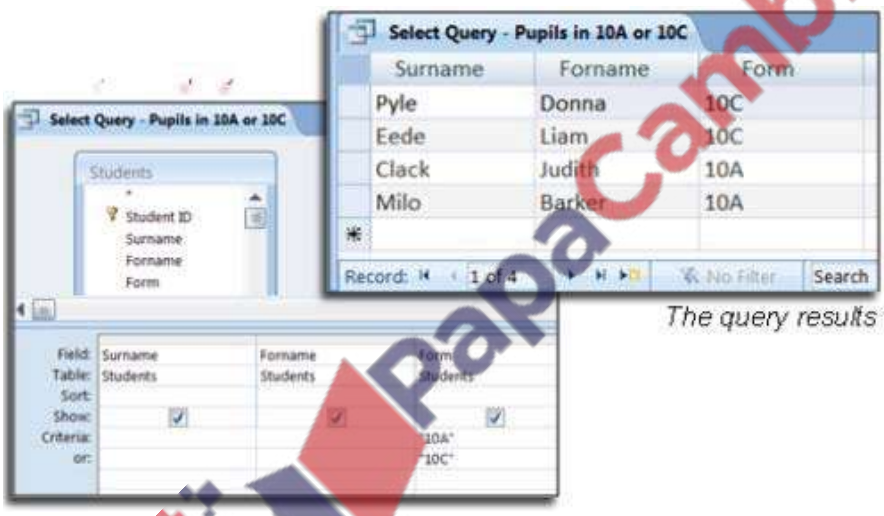
The query design is shown below. Note that this time there are two entries in the search criteria row Also note that this time the query has been given a meaningful name ***("Select Query –***

Pupils in 10B born before 1995"). This saves other database users from unnecessarily creating the same query.



The query results

Below is a new complex query that uses the logical operator OR to find pupils who are in Form 10A or Form 10C: (Form = "10A" OR "Form = "10C") this time, in the query definition there will be two criteria lines. The query and its results are shown below:



The query results

Wildcards in Queries

Wildcard characters can be used in database queries. For example you may want a list of all pupils born in November, or all of the pupils whose surname starts with a 'C'. Wildcard searches allow you to specify the part of the data that you know and leave the data handling software to fill in the blanks.

Surname Like "C*" would find all records where the surname begins with a C.

Quick Revision Questions:

Q 1) Define following terms:

Database: .....

.....

Entity: .....

.....

Table: .....

.....

Field/ Attribute: .....

.....

Record/Tupple: .....

.....

Primary Key: .....

.....

Query: .....

.....

Candidate Example Response Summer 2015 P22

6 A database, MARKS, was set up to record the test results for a class of students. Part of the database is shown below.

Student Name	Class ID	Maths	English	Science	History	Geography
Paul Smith	0017	70	55	65	62	59
Ravi Gupta	0009	29	34	38	41	44
Chin Hwee	0010	43	47	50	45	52
John Jones	0013	37	67	21	28	35
Diana Abur	0001	92	88	95	89	78
Rosanna King	0016	21	13	11	27	15

(a) Give the number of fields that are in each record.

.....[1]

(b) State which fields you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

.....[2]

(c) The query-by-example grid below selects all students with more than 60 marks in History or more than 60 marks in Geography.

Field:	Student Name	History	Geography
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		>60	
or:			>60

Show what would be output.

.....[2]

(d) Complete the query-by-example grid below to select and show the student names only of all students with less than 40 marks in both Maths and English. [3]

Field:			
Table:			
Sort:			
Show:	☐	☐	☐
Criteria:			
or:			

Question 6(a)

Example candidate response – high, middle, low

(a) Give the number of fields that are in each record.

7 [1]

Examiner comment

All but the weakest candidates could identify the number of fields in each record.

Total mark awarded = 1 out of 1

Question 6(b)

Example candidate response – high

(b) State which field you would choose for the primary key.

Class ID

Give a reason for choosing this field.

The Class ID of two students cannot be same as it always unique and so this field can be used to identify a record. [2]

Examiner comment – high

Most candidates could identify the field to choose for a primary key, this candidate gave a good explanation of their choice using appropriate database terminology.

Total mark awarded = 2 out of 2

Example candidate response – middle

(b) State which field you would choose for the primary key.

Class ID

Give a reason for choosing this field.

It is because it is a primary key as it provides with Class ID of students [2]

Examiner comment – middle

Most candidates could identify the field to choose for a primary key; sometimes the explanation did not provide enough information to gain a mark. This explanation just repeats the question and does not add any further information.

Total mark awarded = 1 out of 2

Example candidate response – low

(b) State which field you would choose for the primary key.

Student Name

Give a reason for choosing this field.

It is because it is the main field which gives the general information about student. [2]

Examiner comment – low

Weaker candidates sometimes incorrectly identified the Student Name field; this did not gain a mark.

Total mark awarded = 0 out of 2

Question 6(c)

Example candidate response – high

(c) The query-by-example grid below selects all students with more than 60 marks in History or more than 60 marks in Geography.

Field:	Student Name	History	Geography
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		>60	
or:			>60

Show what would be output.

Diana Abur

Paul Smith [2]

Examiner comment – high

The answer should be the output, this is completely correct as it shows only the Student Names and they are in ascending order.

Total mark awarded = 2 out of 2

Example candidate response – middle

(c) The query-by-example grid below selects all students with more than 60 marks in History or more than 60 marks in Geography.

Field:	Student Name	History	Geography
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		>60	
or:			>60

Show what would be output.

Paul Smith , Diana Abur

Examiner comment – middle

The content of the answer is correct as only the Student Names are shown, the order is incorrect as it is the order the names appear in the database table not in ascending order.

Total mark awarded = 1 out of 2

Example candidate response – low

(c) The query-by-example grid below selects all students with more than 60 marks in History or more than 60 marks in Geography.

Field:	Student Name	History	Geography
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		>60	
or:			>60

Show what would be output.

(Paul Smith > 60 History) ,(Diana Abur > 60 History),
(Diana Abur > 60 Geography). [2]

Examiner comment – low

The candidate appears to know how the query-by-example shown should work, however the question asked has not been answered as the reasoning has been shown rather than the output.

Total mark awarded = 0 out of 2

Example candidate response – high

(d) Complete the query-by-example grid below to select and show the student names only of all students with less than 40 marks in both Maths and English.

Field:	Student Name	Maths	English
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		< 40	< 40
or:			

[3]

Examiner comment – high

The candidate has selected the correct fields. The Sort row for the Student Name can be left blank or set to Ascending or Descending since there are no instructions about sorting. The show boxes are correctly left unchecked for Maths and English. The < 40 criteria for the Maths and English marks are on the same line as both are required.

Total mark awarded = 3 out of 3

Example candidate response – middle

(d) Complete the query-by-example grid below to select and show the student names only of all students with less than 40 marks in both Maths and English.

Field:	Student Name	Maths	English
Table:	MARKS	MARKS	MARKS
Sort:	Descending		
Show:	☑	☐	☐
Criteria:		< 40	
or:			< 40

[3]

Examiner comment – middle

The candidate has selected the correct fields. The Sort row for the Student Name can be left blank or set to Ascending or Descending since there are no instructions about sorting. The show boxes are correctly left unchecked for Maths and English. The < 40 criteria for the Maths and English marks are not on the same line; this is incorrect as both are required. There is no mark for the English column.

Total mark awarded = 2 out of 3

Example candidate response – low

(d) Complete the query-by-example grid below to select and show the student names only of all students with less than 40 marks in both Maths and English.

Field:	Student Name	Maths	English
Table:	Marks	Marks	Marks
Sort:	descending	descending	descending
Show:	☑	☑	☑
Criteria:		<40	
or:			<40

[3]

Examiner comment – low

The candidate has selected the correct fields. The Sort row for the Student Name can be left blank or set to Ascending or Descending since there are no instructions about sorting. The show boxes are incorrectly checked for Maths and English. The < 40 criteria for the Maths and English marks are not on the same line; this is incorrect as both are required. There is no mark for the Maths column and no mark for the English column.

Total mark awarded = 1 out of 3

30 Summer 2019 P22

6 A database table, FLIGHT, is used to keep a record of flights from a small airfield. Planes can carry passengers, freight or both. Some flights are marked as private and only carry passengers.

Flight number	Plane	Notes	Departure time	Passengers
FN101	Caravan 1	Private passenger flight	08:00	Y
CN101	Caravan 2	Freight only	08:30	N
CN102	Piper 1	Freight only	09:00	N
FN104	Piper 2	Passengers only	09:20	Y
FN105	Piper 1	Freight and passengers	10:00	Y
FN106	Caravan 1	Passengers only	10:30	Y
CN108	Caravan 2	Freight only	08:00	N
CN110	Lear	Private passenger flight	08:00	Y

(a) State the field that could have a Boolean data type.

Field ..... [1]

(b) A query-by-example has been written to display just the flight numbers of all planes leaving after 10:00 that only carry passengers.

Field:	Flight number	Passengers	Departure time		
Table:	FLIGHT	FLIGHT	FLIGHT		
Sort:					
Show:	☐	☑			
Criteria:		=Y	= 10:00		
or:					

Explain why the query-by-example is incorrect, and write a correct query-by-example. [7]

Explanation .....

.....

.....

.....

.....

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

Example Candidate Response – high

6 A database table, FLIGHT, is used to keep a record of flights from a small airfield. Planes can carry passengers, freight or both. Some flights are marked as private and only carry passengers.

Flight number	Plane	Notes	Departure time	Passengers
FN101	Caravan 1	Private passenger flight	08:00	Y
CN101	Caravan 2	Freight only	08:30	N
CN102	Piper 1	Freight only	09:00	N
FN104	Piper 2	Passengers only	09:20	Y
FN105	Piper 1	Freight and passengers	10:00	Y
FN106	Caravan 1	Passengers only	10:30	Y
CN108	Caravan 2	Freight only	08:00	N
CN110	Lear	Private passenger flight	08:00	Y

(a) State the field that could have a Boolean data type.

Field Passengers (1) [1]

(b) A query-by-example has been written to display just the flight numbers of all planes leaving after 10:00 that only carry passengers.

Field:	Flight number	Passengers	Departure time	
Table:	FLIGHT	FLIGHT	FLIGHT	
Sort:				
Show:	☐	☑	☐	☐
Criteria:		= Y	= 10:00	
or:				

Explain why the query-by-example is incorrect, and write a correct query-by-example.

Explanation The flight number box has not been ticked which means the flight number will not be displayed. The passengers box has been ticked, however that is not to be displayed. The (2) criteria of the departure time is set to = 10:00, but it should be after 10:00, i.e. >10:00 and only passengers flights have not been specified

Field:	Flight number	Passengers	Departure time	Notes
Table:	FLIGHT	FLIGHT	FLIGHT	(3)
Sort:				
Show:	☑	☐	☐	☐
Criteria:		= Y	>10:00	= "Passengers only"
or:		= Y	>10:00	= "Private passenger flight"

[7]

Examiner Comments

(1) The candidate states the correct field.

Mark for (a) = 1 out of 1

(2) Three errors are identified by the candidate:

- the flight number is not displayed
- the passenger field is displayed when not required
- the criteria for time is incorrect.

(3) This query does not work due to the missing table in the notes field.

Mark for (b) = 6 out of 7

Total mark awarded = 7 out of 8

How the candidate could have improved their answer

The candidate could have completed the table row for each field.

Example Candidate Response – middle

6 A database table, FLIGHT, is used to keep a record of flights from a small airfield. Planes can carry passengers, freight or both. Some flights are marked as private and only carry passengers.

Flight number	Plane	Notes	Departure time	Passengers
FN101	Caravan 1	Private passenger flight	08:00	Y
CN101	Caravan 2	Freight only	08:30	N
CN102	Piper 1	Freight only	09:00	N
FN104	Piper 2	Passengers only	09:20	Y
FN105	Piper 1	Freight and passengers	10:00	Y
FN106	Caravan 1	Passengers only	10:30	Y
CN108	Caravan 2	Freight only	08:00	N
CN110	Lear	Private passenger flight	08:00	Y

(a) State the field that could have a Boolean data type.

Field ...Passengers... 1 [1]

(b) A query-by-example has been written to display just the flight numbers of all planes leaving after 10:00 that only carry passengers.

Field:	Flight number	Passengers	Departure time	
Table:	FLIGHT	FLIGHT	FLIGHT	
Sort:				
Show:	☐	☑	☐	☐
Criteria:		= Y	= 10:00	
or:				

Explain why the query-by-example is incorrect, and write a correct query-by-example.

Explanation ...because the flight number is not displayed as according to the criteria, instead Passengers is shown. Furthermore, the Departure time time should be greater than or equal to 10:00 (not equal to 10:00). Instead of Passengers, Notes should be used as the field. 2

Field:	Flight number	~~Passen~~ Departure time	~~Passengers~~ Notes	3
Table:	FLIGHT	FLIGHT	FLIGHT	
Sort:				
Show:	☑	☐	☐	☐
Criteria:		> = 10:00	= "only Passengers"	
or:				

[7]

Examiner Comments

1 The correct field is chosen by the candidate.

Mark for (a) = 1 out of 1

2 Two errors are identified correctly by the candidate:

- not displaying the flight number
- displaying whether passengers are carried on the flight.

3 The field, table and show rows are correct; the criteria are both incorrect.

Mark for (b) = 4 out of 7

Total mark awarded = 5 out of 8

How the candidate could have improved their answer

The candidate could have completed both the criteria rows, using the correct criteria, > (greater than), for the departure time field and "Passengers only" and "Private passenger flight" for the notes field.

Example Candidate Response – low

6 A database table, FLIGHT, is used to keep a record of flights from a small airfield. Planes can carry passengers, freight or both. Some flights are marked as private and only carry passengers.

Flight number	Plane	Notes	Departure time	Passengers
FN101	Caravan 1	Private passenger flight	08:00	Y
CN101	Caravan 2	Freight only	08:30	N
CN102	Piper 1	Freight only	09:00	N
FN104	Piper 2	Passengers only	09:20	Y
FN105	Piper 1	Freight and passengers	10:00	Y
FN106	Caravan 1	Passengers only	10:30	Y
CN108	Caravan 2	Freight only	08:00	N
CN110	Lear	Private passenger flight	08:00	Y

(a) State the field that could have a Boolean data type.

Field ..... Passengers. 1 ..... [1]

(b) A query-by-example has been written to display just the flight numbers of all planes leaving after 10:00 that only carry passengers.

Field:	Flight number	Passengers	Departure time	
Table:	FLIGHT	FLIGHT	FLIGHT	
Sort:				
Show:	☐	☑	☐	☐
Criteria:		= Y	= 10:00	
or:				

Explain why the query-by-example is incorrect, and write a correct query-by-example.

Explanation ..It is incorrect because it is not showing flight number 2 and Departure time mast be 10:00 or after but it only shows 10:00.

Field:	Flight number	Passenger	Departure time 3	
Table:	FLIGHT	FLIGHT	FLIGHT	
Sort:				
Show:	☑	☐	☐	☐
Criteria:		=Y	>= 10:00	
or:				

[7]

Examiner Comments

1 The correct field is identified by the candidate.

Mark for (a) = 1 out of 1

2 One error – not showing the flight number – is identified correctly by the candidate. However, the statement about the departure time is incorrect.

3 The show row is correct. However, there is a field missing and the criteria row is incorrect.

Mark for (b) = 2 out of 7

Total mark awarded = 3 out of 8

How the candidate could have improved their answer

The candidate could have identified more than one error correctly and used the required fields and correctly completed both criteria rows.

Common mistakes candidates made in this question

Candidates didn't always realise that flights that carried only passengers were required and included flights that carried passengers and freight.

7 A database ELEMENTS was set up to show the properties of certain chemical elements. Part of the database is shown below.

Name of element	Element symbol	Atomic number	Atomic weight	Melting point (C)	Boiling point (C)	State at room temp
oxygen	O	8	16	–218	–183	gas
iron	Fe	26	56	1538	2861	solid
mercury	Hg	80	201	–38	356	liquid
bromine	Br	35	80	–7	59	liquid
osmium	Os	76	190	3033	5012	solid
caesium	Cs	55	133	28	671	solid
gallium	Ga	31	70	30	2204	solid
argon	Ar	18	40	–189	–186	gas
silver	Ag	47	108	961	2162	solid

(a) How many fields are in each record?

..... [1]

(b) The following search condition was entered:

(Melting point (C) < 40) AND (Atomic weight > 100)

Using Element symbol only, which records would be output?

.....

..... [2]

(c) Which field would be best suited as primary key?

.....[1]

14.10 Summer 2015 P21& 23

7 A database, PROPERTY, was set up to show the prices of properties for sale and the features of each property. Part of the database is shown below.

Property Type	Brochure No	Number of Bedrooms	Number of Bathrooms	Garden	Garage	Price in $
Bungalow	B17	7	4	Yes	Yes	750,000
Apartment	A09	2	1	No	No	100,000
House	H10	4	2	Yes	No	450,000
House	H13	3	2	Yes	No	399000
Apartment	A01	2	2	No	Yes	95000
Apartment	A16	1	1	No	No	150000
House	H23	3	1	No	Yes	250000
House	H46	2	1	Yes	Yes	175000

(a) Give the number of fields that are in each record.

.....[1]

(b) State which field you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

.....[2]

(c) State the data type you would choose for each of the following fields.

Garage .....

Number of Bedrooms .....

Price in $ .....[3]

(d) The query-by-example grid below selects all houses with more than 1 bathroom and more than 2 bedrooms.

Field:	Property Type	Number of Bedrooms	Number of Bathrooms	Price in $	Brochure No
Table:	PROPERTY	PROPERTY	PROPERTY	PROPERTY	PROPERTY
Sort:				Ascending	
Show:	☐	☐	☐	☑	☑
Criteria:	= 'House'	>2	>1		
or:					

Show what would be output.

.....

.....[2]

(e) Complete the query-by-example grid below to select and show the brochure number, property type and price of all properties with a garage below $200,000.

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

[4]

Examiner Report Question 7
(a) Many candidates correctly identified the number of fields in each record.
(b) Most candidates correctly identified the field to choose for the primary key. Better candidates gave a correct reason for their choice.
(c) Nearly all candidates correctly stated at least one data type.
(d) Most candidates correctly showed only the Price in $ and the Brochure No, as identified by the query-by-example grid. Better candidates showed attention to detail, by correctly putting the prices in ascending order and the Price in $ field before the Brochure No field as indicated by the query-by-example grid.
(e) Most candidates correctly identified the fields to include in the query-by-example grid and identified those that were to be shown. A common error was to incorrectly set the criterion for the garage, when the data type had been set as a Boolean field in part (c).

14.11 Summer 2015 P22

6 A database, MARKS, was set up to record the test results for a class of students. Part of the database is shown below.

Student Name	Class ID	Maths	English	Science	History	Geography
Paul Smith	0017	70	55	65	62	59
Ravi Gupta	0009	29	34	38	41	44
Chin Hwee	0010	43	47	50	45	52
John Jones	0013	37	67	21	28	35
Diana Abur	0001	92	88	95	89	78
Rosanna King	0016	21	13	11	27	15

(a) Give the number of fields that are in each record.

.....[1]

(b) State which fields you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

.....[2]

(c) The query-by-example grid below selects all students with more than 60 marks in History or more than 60 marks in Geography.

Field:	Student Name	History	Geography
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	✓	☐	☐
Criteria:		>60	
or:			>60

Show what would be output.

.....

.....[2]

(d) Complete the query-by-example grid below to select and show the student names only of all students with less than 40 marks in both Maths and English. [3]

Field:			
Table:			
Sort:			
Show:	☐	☐	☐
Criteria:			
or:			

Examiner's comments on Question 6

(a) Many candidates correctly identified the number of fields in each record.

(b) Most candidates correctly identified the field to choose for the primary key. Better candidates gave a correct reason for their choice.

(c) Better candidates correctly showed only the student names as identified by the query-by-example grid. Some of these candidates correctly ordered the names in ascending order.

(d) Most candidates correctly identified the fields to include in the query-by-example grid and identified those that were to be shown. A common error was to set the Maths or English criteria to OR rather than AND, where both criteria are on the same row.

14.12 Winter 2015 P21 & 22

6 A picture gallery owner has decided to set up a database to keep information about the pictures he has for sale. The database table, PICTURE, will contain the following fields:
Title; Artist; Description; Catalogue Number; Size (area in square centimetres); Price; Arrived (date picture arrived at gallery); Sold (whether picture is already sold)

(a) (i) State what data type you would choose for each field.

Title .....

Artist .....

Description .....

Catalogue Number .....

Size .....

Price .....

Arrived .....

Sold .....[4]

(ii) State which field you would choose for the primary key.

.....[1]

(b) Give a validation check that you can perform on each of these fields. Each validation check must be different.

Catalogue Number .....

Size .....

Price .....

Arrived .....[4]

(c) Complete the query-by-example grid below to select and show the Catalogue Number, Title and Price of all unsold pictures by the artist 'Twister'.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

Examiners' Comments Question 6

(a) (i) Most candidates correctly identified the correct data type for some of the fields. Candidates who did less well throughout, incorrectly used data types from programming rather than database management.
(ii) Most candidates correctly identified the field to choose for the primary key.
(b) Many candidates correctly identified at least one suitable validation check. Candidates with stronger responses throughout identified four different checks; a few candidates incorrectly repeated a validation check.
(c) Many candidates correctly identified the fields to include in the query-by-example grid; stronger responses identified those fields that were to be shown. A common error was to not include the table name.

14.13 Winter 2015 P23

5 A motor boat hire company decides to set up a database to keep information about boats that are available for hire. The database table, BOAT, will contain the following fields:

Boat Name; Model; Engine Power (in hp); Number of Seats; Life Raft (whether there is a life raft kept on the boat); Day Price (price for a day's hire).

(a) Give the data type you would choose for each field.

Boat Name .....

Model .....

Engine Power .....

Number of Seats .....

Life Raft .....

Day Price .....[3]

(b) State a validation check that you can perform on each of these fields. Each validation check must be different.

Boat Name .....

Model .....

Number of Seats .....

Day Price .....[4]

(c) Complete the query-by-example grid below to select and show the Boat Name, Model and Day Price of a day's hire for all boats with 4 seats and an Engine Power of more than 100 hp.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

14.14 Summer 2016 P21 &P23

6 A database, STAFFPHONE, was set up to show the telephone extension numbers for members of staff working in a department store.

Name	Department	Extension number
Jane Smith	Toys	129
Sue Wong	Books	124
David Chow	Toys	129
Amy Tang	Household	123
Joe Higgs	Books	124
Jane Smith	Shoes	125
Adel Abur	Shoes	125
Peter Patel	Toys	129

(a) Explain why none of the fields in the database can be used as a primary key.

.....

.....

.....[2]

(b) State a field that could be added as a primary key.

.....

Give a reason for choosing this field.

.....

..... [2]

(c) Use the query-by-example grid below to provide a list of all members of staff, in alphabetical order, grouped by department. [5]

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

14.15 Summer 2016 P22

7 A database, SOFASELECT, was set up to show the prices of suites, sofas and chairs for sale from an online furniture warehouse. Part of the database is shown below.

Description	Brochure Number	Number of Seats	Number of Pieces	Material	Colour	Price in $
Sofa	SF17	2	1	Leather	Red	950
Sofa	SF19	3	1	Vinyl	Black	1,000
Suite	SU10	4	3	Velvet	Green	1,500
Suite	SU23	5	3	Leather	Brown	950
Recliner chair	RC01	1	1	Leather	Cream	600
Chair	CH16	1	1	Vinyl	Red	250
Recliner sofa	RS23	4	1	Leather	Cream	1,200
Chair	CH10	1	1	Velvet	Red	175

(a) How many fields are in each record?

.....[1]

(b) State which field you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

.....[2]

(c) State the data type you would choose for each of the following fields.

Number of Seats .....

Price in $ .....[2]

(d) The query-by-example grid below selects all the furniture in cream leather.

Field:	Description	Material	Colour	Price in $	Brochure Number
Table:	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT	SOFASELECT
Sort:				Descending	
Show:	☑	☐	☐	☑	☑
Criteria:		= 'Leather'	= 'Cream'		
or:					

Show the output from the query-by-example.

.....

.....[3]

(e) Complete the query-by-example grid below to select and show the brochure number, material, colour and price of all the furniture with 3 or more seats.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

14.16 Winter 2016 P21-23

6 A database, THEATRETOURS, was set up to show the tour dates, towns, number of seats and prices in local currency for a Shakespeare play.

Town	Tour Date	Number of Seats	Price Local Currency
Wigan	18/08/2016	120	15.00
Dumfries	20/08/2016	160	12.50
Turin	25/08/2016	200	17.00
Macon	27/08/2016	75	18.00
Bordeaux	29/08/2016	170	20.00
Algiers	01/09/2016	125	1350.00
Windhoek	05/09/2016	65	90.00
Windhoek	06/09/2016	65	90.00
Port Elizabeth	10/09/2016	200	110.00

(a) Explain why none of the fields in the database can be used as a primary key.

.....

.....

.....[2]

(b) State a field that could be added as a primary key.....

Give a reason for choosing this field.....

.....[2]

(c) Use the query-by-example grid below to provide a list of tour dates and seat prices in alphabetical order of town. [4]

Field					
Table					
Sort					
Show:					
Criteria					
or					

14.17 Winter 2016 P22

5 A database, PLAYPRODUCTION, was set up to show the performance dates, prices and number of seats available at a theatre specialising in Shakespeare productions.

Play	**Performance Date**	**Number Seats Stalls**	**Number Seats Circle**	**Price Stalls Seats $**	**Price Circle Seats $**
As You Like It	01/07/2016	120	90	20.00	30.00
As You Like It	02/07/2016	85	45	30.00	40.00
As You Like It	09/07/2016	31	4	30.00	40.00
Macbeth	14/07/2016	101	56	25.00	35.00
Macbeth	15/07/2016	50	34	25.00	35.00
Macbeth	16/07/2016	12	5	35.00	50.00
Julius Caesar	22/07/2016	67	111	20.00	20.00
Julius Caesar	23/07/2016	21	24	15.00	15.00
A Comedy of Errors	30/07/2016	45	36	35.00	45.00

(a) Give the number of fields that are in each record.

..... [1]

(b) State the data type you would choose for each of the following fields.

Play .....

Number Seats Stalls .....

Price Stalls Seats $ ..... [3]

(c) The query-by-example grid below selects all the productions with more than 100 seats left in either the stalls or the circle.

Field:	Play	Performance Date	Number Seats Stalls	Number Seats Circle
Table:	PLAYPRODUCTION	PLAYPRODUCTION	PLAYPRODUCTION	PLAYPRODUCTION
Sort:	Ascending			
Show:	☑	☑	☐	☐
Criteria:			> 100	
or:				> 100

Show what would be output from the query-by-example.

.....

.....

..... [3]

(d) Complete the query-by-example grid below to select all the productions with at least six seats left in the circle and show the Play, Performance Date and Price Circle Seats $ in Performance Date order. [5]

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

14.18 March 2017 P21 (India)

6 A database table, DEVICE, has been set up to record the electronic equipment used in a small business.

Device ID	Device Type	User	Purchase Date	Purchase Price ($)	Portable
3	Desktop	Alan Swales	14/02/2017	1350.00	N
4	Laptop	Chantel Law	01/02/2016	1460.00	Y
5	Tablet	Abdula Saud	31/12/2016	1000.00	Y
6	Desktop	Abdula Saud	14/03/2017	1000.00	N
7	Laptop	Alan Swales	15/03/2016	1700.00	Y
8	Tablet	TaonaJaji	16/12/2016	470.00	Y

(a) The query-by-example grid below selects certain records.

Field:	User	Portable	Purchase Price ($)
Table:	DEVICE	DEVICE	DEVICE
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		Y	>1000
or:			

Show what would be the output from the query-by-example.

.....

.....

.....[2]

(b) Complete the query-by-example grid below to select all Desktop devices that were either purchased before 31/12/2016 or cost under $1000. Only show the Device ID and DeviceType.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

14.19 Summer 2017 P21

7 A television (TV) store has a database table, TVSTOCK, for its new range of televisions. The table stores the screen size of each TV, whether it will show 3D, whether the screen is curved or flat, if the internet is available on the TV, if it has a built-in hard disk drive and the price. Part of the database table is shown below.

TVID	ScreenSize	3D	CurvedFlat	Internet	HDD	Price
TV80CVINT	80	YES	CV	YES	YES	$7,000.00
TV65CVINT	65	YES	CV	YES	YES	$5,000.00
TV60CVINT	60	YES	CV	YES	YES	$4,500.00
TV60FTINT	60	YES	FT	YES	YES	$4,000.00
TV55CVINT	55	YES	CV	YES	NO	$3,000.00
TV55FTINT	55	YES	FT	YES	NO	$3,500.00
TV55FTNIN	55	YES	FT	NO	NO	$3,000.00
TV50CVINT	50	YES	CV	YES	NO	$2,500.00
TV50FTINT	50	YES	FT	YES	NO	$2,000.00
TV50FTNIN	50	YES	FT	NO	NO	$1,750.00
TV42FTINT	42	YES	FT	YES	NO	$1,500.00
TV37FTINT	37	NO	FT	YES	NO	$1,200.00
TV20FTNIN	20	NO	FT	NO	NO	$800.00
TV15FTNIN	15	NO	FT	NO	NO	$400.00

(a) State the type of the field **TVID** and give a reason for your choice.

.....

..... [1]

(b) Complete the table with the most appropriate data type for each field. [3]

Field name	Data type
ScreenSize	
3D	
CurvedFlat	
Internet	
HDD	
Price	

(c) Use the query-by-example grid below to provide a list of all of the curved screen TVs that have a built-in hard disk drive. Make sure the list only displays the TVID, the price and the screen size in ascending order of price. [5]

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

14.20Summer 2017 P22

5 A database table, SHEEP, is used to keep a record of the sheep on a farm. Each sheep has a unique ear tag, EARnnnn; n is a single digit. The farmer keeps a record of the date of birth, the gender and the current weight of each sheep in kilograms.

(a) Identify the **four** fields required for the database. Give each field a suitable name and data type. Provide a sample of data that you could expect to see in the field. [8]

Field 1 name.....

Data type .....

Data sample .....

Field 2 name.....

Data type .....

Data sample .....

Field 3 name.....

Data type .....

Data sample .....

Field 4 name.....

Data type .....

Data sample .....

(b) State the field that you would choose as the primary key.

.....[1]

(c) Using the query-by-example grid below, write a query to identify the ear tags of all male sheep weighing over 10 kilograms. Only display the ear tags. [3]

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

14.21 Winter 2017 P21

6 A wildlife park has a database table, called LIVESTOCK, to classify and record its animal species. Part of the database table is shown.

Species	Classification	Diet	Legs
Giraffe	Mammal	Herbivore	4
Elephant	Mammal	Herbivore	4
Crocodile	Reptile	Carnivore	4
Ostrich	Bird	Omnivore	2
Gorilla	Mammal	Herbivore	2
Bear	Mammal	Omnivore	4
Rhinoceros	Mammal	Herbivore	4
Hippopotamus	Mammal	Herbivore	4
Flamingo	Bird	Omnivore	2
Lion	Mammal	Carnivore	4
Turtle	Reptile	Omnivore	4
Penguin	Bird	Carnivore	2

(a) Suggest another appropriate field that could be added to this database by stating its name and data type. State its purpose and give an example of the data it could contain.

Field name .....

Data Type .....

Purpose .....

.....

Example of data .....[2]

(b) Use the query-by-example grid below to provide a list of all four legged mammals that are herbivores, sorted alphabetically by species, with only the species displayed.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

14.22 Winter 2017 P22

6 A database table, TRAIN, is to be set up for a railway company to keep a record of the engines available for use. Each engine has a unique number made up of 5 digits, nnnnn. The engines are classified as freight (F) or passenger (P) together with a power classification that is a whole number between 0 and 9, for example F8. The railway company keeps a record of the date of the last service for each engine.

(a) Identify the **three** fields required for the database. Give each field a suitable name and data type. Provide a sample of data that you could expect to see in the field.

Field 1 Name .....

Data type .....

Data sample .....

Field 2 Name .....

Data type .....

Data sample .....

Field 3 Name .....

Data type .....

Data sample .....[6]

(b) State the field that you should choose as the primary key.

.....[1]

(c) Using the query-by-example grid below, write a query to identify all passenger engines that have not been serviced in the past 12 months. Only display the engine numbers. [3]

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

14.23 March 2018 P22 (India)

6 A database table, JEWEL, is used to keep a record of jewellery for sale in a shop. Each item of jewellery can be made of silver, platinum or gold metal. The shop stocks rings, bracelets and necklaces. The number in stock and the price is also stored.

(a) Identify the **four** fields required for the database. Give each field a suitable name and data type. Explain why you chose the data type for each field.

Field 1 Name ..... Data type .....

Explanation .....

.....

Field 2 Name ..... Data type .....

Explanation .....

.....

Field 3 Name ..... Data type .....

Explanation .....

.....

Field 4 Name ..... Data type .....

Explanation .....

.....[8]

(b) Explain why none of these fields could be used as a primary key.

.....

.....[1]

(c) Using the query-by-example grid below, write a query to identify the silver bracelets. Only display the number in stock and the price.

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

Comments on Question 6

(a) *Nearly all candidates identified some appropriate fields and could also provide a suitable data type and explanation. Many candidates provided excellent answers worth full marks.*

(b) *Nearly all candidates gave a correct explanation as to why none of the fields were suitable use as a primary key.*

(c) *Nearly all candidates correctly identified the fields required in the query-by-example grid. Most candidates correctly identified which fields to show. Many candidates provided suitable criteria to identify that only details of silver bracelets were required.*

14.24Summer 2018 P21

6 A database table, PERFORMANCE, is used to keep a record of the performances at a local theatre.

Show Number	Type	Title	Date	Sold Out
SN091	Comedy	An Evening at Home	01 Sept	Yes
SN102	Drama	Old Places	02 Oct	No
SN113	Jazz	Acoustic Evening	03 Nov	No
SN124	Classical	Mozart Evening	04 Dec	Yes
SN021	Classical	Bach Favourites	01 Feb	Yes
SN032	Jazz	30 Years of Jazz	02 Mar	Yes
SN043	Comedy	Street Night	03 Apr	No
SN054	Comedy	Hoot	04 May	No

(a) State the number of fields and records in the table.

Fields .....

Records .....[2]

(b) Give **two** validation checks that could be performed on the **Show Number** field.

Validation check 1 .....

.....

Validation check 2 .....

..... [2]

(c) Using the query-by-example grid, write a query to identify jazz performances that are not sold out. Only display the date and the title. [4]

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

14.25Summer 2018 P21

6 A database table, TREES, is used to keep a record of the trees in a park. Each tree is given a unique number and is examined to see if it is at risk of dying. There are over 900 trees; part of the database table is shown.

Tree Number	Type	Map Position	Age in Years	At Risk
TN091	Acacia	A7	250	Y
TN172	Olive	C5	110	N
TN913	Cedar	B9	8	N
TN824	Banyan	A3	50	Y
TN021	Pine	D5	560	Y
TN532	Teak	C8	76	Y
TN043	Yew	B1	340	N
TN354	Spruce	D4	65	N
TN731	Elm	B10	22	Y
TN869	Oak	C9	13	N
TN954	Pine	E11	3	N

(a) State the number of fields in the table.

.....[1]

(b) The tree numbering system uses TN followed by three digits. The numbering system will not work if there are over 1000 trees.

Describe, with the aid of an example, how you could change the tree numbering system to allow for over 1000 trees. Existing tree numbers must not be changed.

.....

.....

.....

.....[2]

(c) Using the query-by-example grid, write a query to identify at risk trees over 100 years old. Display only the type and the position on the map. [4]

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

14.26 Winter 2018 P21

6 A database table, PORTRAIT, is used to keep a record of the portraits available from a photographic studio. Each portrait has a unique reference number PICnnn, where n is a single digit, for examplePIC123. The studio keeps a record of the size (for example 20 × 15), the type (black and white or colour), and the price in dollars.

(a) Complete the table to show the most appropriate data type for each of the fields.[4]

Field	**Data type**
Reference Number	
Size	
Type	
Price in $	

(b) The results from the query-by-example grid should show the reference number, price, type and size of all portraits under $50. Identify the **three** errors in the query-by-example grid.

Field:	Reference No	Price in $	Type	Size
Table:	PORTRAIT	PORTRAIT	PORTRAIT	PORTRAIT
Sort:				
Show:	☑	☑	☐	☑
Criteria:		>50.00		
or:				

Error 1 .....

.....

Error 2 .....

.....

Error 3 .....

.....[3]

14.27 Winter 2018 P23

6 An online fruit tree specialist sells fruit trees in various sizes. A database table, TREETAB, shows the tree type and, for each size, the price and whether they are in stock.

Tree Type	Size1	Size1 In	Size2	Size2 In	Size3	Size3 In
Apple	10.95	Yes	14.95	Yes	29.95	Yes
Apple	12.95	Yes	14.95	Yes	29.95	Yes
Cherry	24.95	No	34.95	No	59.95	Yes
Fig	19.95	Yes	29.95	No	49.95	Yes
Guava	19.95	No	29.95	No	59.95	No
Nectarine	8.50	Yes	11.95	Yes	19.95	Yes
Olive	19.95	No	39.95	Yes	59.95	Yes
Peach	9.25	No	1.95	Yes	19.95	Yes
Pear	10.95	Yes	14.95	Yes	29.95	Yes
Plum	8.95	Yes	11.95	Yes	19.95	Yes
Pomegranate	12.95	No	18.95	Yes	34.95	No
Quince	34.95	Yes	44.95	Yes	84.95	No

(a) State whether any of the fields shown would be suitable as a primary key.

.....

Explain your answer

.....

.....

.....[2] **(**

b) Complete the table to show the most appropriate data type for each of the fields based on the data shown in the table at the start of question 6. [3]

Field	Data type
Tree Type	
Size3	
Size2 In	

(a) Show the output that would be given by this query-by-example.

Field:	Tree Type	Size 1	Size 1 In	
Table:	TREETAB	TREETAB	TREETAB	
Sort:		Descending		
Show:	☑	☑	☑	☐
Criteria:		<10.00		
or:				

.....

.....

.....

.....[4]

(d) Using the following query-by-example grid, write a query to identify all types of the fruit trees that are out of stock for all three sizes. Make sure the type of the tree and the various 'in stock' fields are shown. The trees should be listed in alphabetical order by type.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

14.28 March 2019 P22

6 A database table, BIKETYRES, is used to keep a record of tyres for sale in a cycle shop.

Tyres are categorised by width and diameter in milimetres, whether they have an inner tube and the type of terrain for which they are designed.

Tyre Code	Width	Diameter	Tube	Terrain	Stock Level
SLTT	23	700	YES	Asphalt	18
MLNT	24	700	NO	Asphalt	23
LLNT	28	700	NO	Asphalt	19
SLTM	23	700	YES	Mixed	22
MLTM	24	700	YES	Mixed	14
LLTM	28	700	YES	Mixed	12
SLTH	23	700	YES	Hard	10
MLTH	24	700	YES	Hard	5
LLNH	28	700	NO	Hard	7
SLNM	23	700	NO	Mixed	12
MLNM	24	700	NO	Mixed	22
LLNM	28	700	NO	Mixed	18
SSNT	23	650	NO	Asphalt	10
MSNT	24	650	NO	Asphalt	8
SSTM	23	650	YES	Mixed	5
MSNM	24	650	NO	Mixed	4

The query-by-example grid below displays the tyre code and the stock level of all 28mm width tyres suitable for mixed terrain.

Field:	Tyre Code	Stock Level	Width	Terrain
Table:	BYKETYRES	BYKETYRES	BYKETYRES	BYKETYRES
Sort:				
Show:	☑	☑	☐	☐
Criteria:			=28	='Mixed'
or:				

Alter the query to show the tyre code and stock level in ascending order of stock level for all 24mm asphalt terrain tyres. Write the new query in the following query-by-example grid.

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

14.29 Summer 2019 P21

5 The table, BEVERAGES, shows the number of calories in 100ml of a range of popular beverages. It also shows the availability of these drinks in a can, a small bottle and a large bottle.

BevNo	BevName	Calories	Can	Small Bottle	Large Bottle
Bev01	Cola	40	Yes	Yes	Yes
Bev02	Lime	45	Yes	No	Yes
Bev03	Energy Drink 1	52	Yes	Yes	No
Bev04	Energy Drink 2	43	Yes	No	No
Bev05	Mango	47	Yes	No	Yes
Bev06	Lemon Iced Tea	38	Yes	No	Yes
Bev07	Lemonade	58	Yes	Yes	Yes
Bev08	Orange Juice	46	Yes	Yes	No
Bev12	Apple Juice	50	Yes	Yes	No
Bev15	Chocolate Milk	83	Yes	Yes	No

(a) Give a reason for choosing BevNo as the primary key for this table.

.....

..... [1]

(b) State the number of records shown in the table BEVERAGES.

..... [1]

(c) List the output that would be given by this query-by-example.

Field:	BevNo	BevName	Can	Small Bottle	Large Bottle
Table:	BEVERAGES	BEVERAGES	BEVERAGES	BEVERAGES	BEVERAGES
Sort:		Descending			
Show:	☑	☑	☐	☐	☐
Criteria:			= "Yes"	= "Yes"	= "Yes"
or:					

.....

.....

.....

..... [3]

(d) Complete the query-by-example grid to output a list showing just the names and primary keys of all the beverages with a calorie count greater than 45. The list should be in alphabetical order of names. [4]

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

14.30 Summer 2019 P22

6 A database table, FLIGHT, is used to keep a record of flights from a small airfield. Planes can carry passengers, freight or both. Some flights are marked as private and only carry passengers.

Flight number	Plane	Notes	Departure time	Passengers
FN101	Caravan 1	Private passenger flight	08:00	Y
CN101	Caravan 2	Freight only	08:30	N
CN102	Piper 1	Freight only	09:00	N
FN104	Piper 2	Passengers only	09:20	Y
FN105	Piper 1	Freight and passengers	10:00	Y
FN106	Caravan 1	Passengers only	10:30	Y
CN108	Caravan 2	Freight only	08:00	N
CN110	Lear	Private passenger flight	08:00	Y

(a) State the field that could have a Boolean data type.

Field ..... [1]

(b) A query-by-example has been written to display just the flight numbers of all planes leaving after 10:00 that only carry passengers.

Field:	Flight number	Passengers	Departure time		
Table:	FLIGHT	FLIGHT	FLIGHT		
Sort:					
Show:	☐	☑	☐		
Criteria:		=Y	= 10:00		
or:					

Explain why the query-by-example is incorrect, and write a correct query-by-example. [7]

Explanation .....

.....

.....

.....

.....

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

Introduction to Logo

LOGO is a programming language that was developed at the MIT Artificial Intelligence Lab in the 1970s. It was designed to be used as an introduction for people to both programming and artificial intelligence. However, while it is easy to learn, Logo is a powerful language. Once you know the basics, Logo can be used to do extremely complicated things. Logo code has been used in telecommunications, multimedia software and robotics. Finally Logo is FUN!

Turtle

When Logo started, the Turtle was a robotic creature which sat on the floor and was directed to move around by a user typing commands on a computer. Today the turtle is represented by an icon on the screen and can be made to draw on the screen using the same commands. In some environments the turtle looks like a turtle (with head, tail and feet) in others represent the turtle with a triangle.

Logo Commands

The following are some of the most use full commands in the Logo Language, which you will want to become familiar with:

FORWARD - Follow this command with a number (such as: 10 or 1000.) A small number will cause the turtle to move forward a short distance. A larger number will cause it to move further. If you select a large enough number the turtle will go off the canvas and wrap around to the other side.

BACK- Follow this command with a number, the same as FORWARD, only this time the turtle will move backwards.

RIGHT - Follow this command with a number between 0 and 360. The turtle will turn right specified number of degrees.

LEFT - Follow this command with a number between 0 and 360. This command is the same as RIGHT only it will turn the turtle left, not right.

PENUP - This command will cause the turtle to pick up its “pen” up so that you can move the turtle without drawing a line.

PENDOWN - This is the command you would use to put the “pen” back down so you can draw again.

SETPENCOLOR - You can change the colour your turtle draws in. Follow the command with a number to get different colours. For example

“SETPENCOLOR 0” would give you a black pen.

CLEAN- This command will erase the canvas

HOME- This command will move the turtle back to the centre of the canvas

The Repeat Command

You can get your turtle to do one (or several) things repeatedly, without typing them again and again using the REPEAT command. Typing

```
REPEAT 4 [FORWARD 10]
```

Would cause the turtle to move forward 10 spaces, 4 time. So, in total the turtle would move forward 40 spaces. Now Try These. Type:

```
REPEAT 4 [FORWARD 50 RIGHT 90]
```

You should get a square. For a bigger square try replacing 50 with 100 or 200.Type:

```
REPEAT 360 [FORWARD 2 LEFT 1]
```

You should get a circle. For a smaller circle try replacing 2 with 1. Can you make a bigger one?

Your First Program

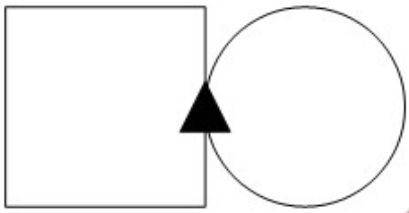
That's a lot to type every time you want to make a square or circle though. Can it be easier? YES. You can teach Logo what a square (or a circle, or a flower) is by making it program. Try typing:

```
TO SQUARE
REPEAT 360 [FORWARD 80 LEFT 90]
END
```

Now type SQUARE and see what happens.

How would you write the program CIRCLE?

How would you write the program CIRCLE_AND_SQUARE to make a drawing that looks like this (where the black triangle is the turtle at the end)?



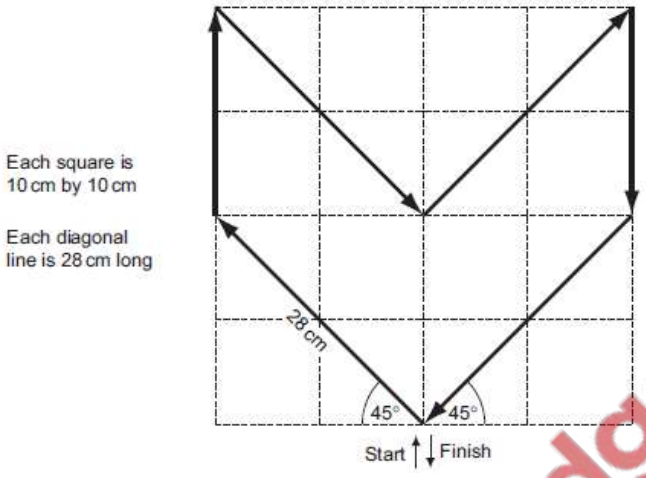
```
TO CIRCLE_AND_SQUARE
HOME
CLEAN
CIRCLE
FORWARD 52
REPEAT 3 [ LEFT 90 FORWARD 104]
LEFT 90
FORWARD 52
END
```

CIRCLE was defined as:

```
TO CIRCLE
REPEAT 360 [FORWARD 1 RIGHT 1]
END
```

15.1 Summer 2014 P11

A floor turtle can use the following instructions.



Complete the set of instructions to draw the above shape.

Pen Down ……

Left 45 ……

……

……

……

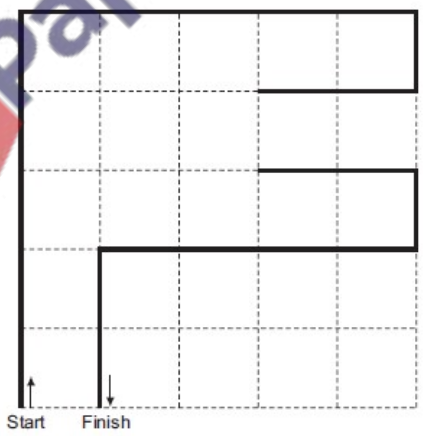
……

……

15.2 Summer 2014

A floor turtle uses the following commands:

In the following grid, each of the squares measures 10 cm by 10 cm:



Complete the set of instructions to draw the shape shown above (in bold lines).

Pen Down ……

Repeat 2……

……

……

……

……

……

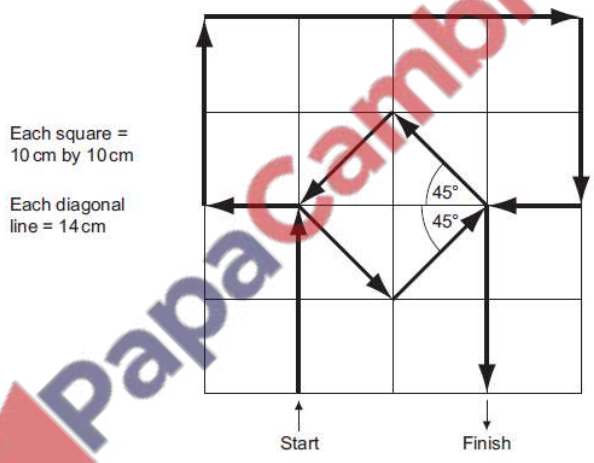
……

1 mark for each block of code between dotted lines. (e.g. use of REPEAT and BACKWARD functions). If an error occurs in the code, try to find a correct code sequence later on in the answer (in cases such as this, it is often easier to work backwards from last statement looking for correct blocks).

15.3 Summer 2012

A floor turtle can use the following instructions.

Instruction	Meaning
FORWARD ***x***	Move ***x*** cm forwards
LEFT ***t***	Turn left ***t*** degrees
RIGHT ***t***	Turn right ***t*** degrees
REPEAT ***n***	Repeat next set of instructions ***n*** times
ENDREPEAT	Finish repeated instructions
PENUP	Lift the pen
PENDOWN	Lower the pen



Complete the set of instructions to draw the above shape.

Pen Down .....

Forward 20.....

Left 90 .....

.....

.....

.....

15.4 Summer 2010

A floor turtle can use the following instructions:

Instruction	Meaning
FORWARD ***d***	Move ***d*** cm forward
BACKWARD ***d***	Move ***d*** cm backward
LEFT ***t***	Turn left ***t*** degrees
RIGHT ***t***	Turn right ***t*** degrees
REPEAT ***n***	Repeat the next set of instructions ***n*** times
ENDREPEAT	End of REPEAT loop
PENUP	Raise the pen
PENDOWN	Lower the pen

(In the following grid, each square is 10 cm by 10 cm.)

Complete the set of instructions to draw the above shape.

Pen Down .....

Left 90 .....

Forward 10.....

Right 90 .....

.....

.....

.....

.....

Marking Scheme

15.1 Summer 2014 P11

pendown
left 45
forward 28
right 45 (1 mark)
forward 20
right 135 (1 mark)
forward 28
left 90 (1 mark)
forward 28
right 135 (1 mark)
forward 20
right 45 (1 mark)
forward 28
(penup)
(left 45)

15.2 Summer 2014

PENDOWN
REPEAT 2
3 FORWARD 50 FORWARD 25 ENDREPEAT
4 RIGHT 90 RIGHT 90 FORWARD 50
5 ENDREPEAT RIGHT 90
- -
6 FORWARD 10
7 RIGHT 90
8 FORWARD 20
- - - - - - - - - - - - - - - - -
9 PENUP (statements 9 and 10 are interchangeable)
10 LEFT 90
11 FORWARD 10
- - - - - - - - - - - - - - - - -
12 PENDOWN
13 LEFT 90 (statements 12 and 13 are interchangeable)
14 FORWARD 20
15 RIGHT 90
- - - - - - - - - - - - - - - - -
16 FORWARD 10
17 RIGHT 90
18 FORWARD 40
- - - - - - - - - - - - - - - - -
19 LEFT 90
20 FORWARD 20
(21 PENUP) (line 21 is not essential) [6

15.3 Summer 2012

pendown
forward 20
left 90
- - - - - - - - - - - -
forward 10
right 90
forward 20 (1 mark)
- - - - - - - - - - - -
right 90
forward 40
right 90
forward 20 (1 mark)
right 90
- - - - - - - - - - - -
forward 10
right 45
forward 14 (1 mark)
- - - - - - - - - - - -
repeat 3 or left 90
left 90 or forward 14
forward 14 or left 90
endrepeat or forward 14 (1 mark)
left 90
forward 14
- - - - - - - - - - - -
right 135
forward 20
(PENUP) (1 mark)

15.4 Summer 2010

LEFT 90
PENDOWN
FORWARD 10
RIGHT 90

FORWARD 10
PENUP
FORWARD 10
PENDOWN

FORWARD 20
RIGHT 90
FORWARD 20
RIGHT 90
FORWARD 20

LEFT 90
FORWARD 20
PENUP / RIGHT 90
---

20 RIGHT 90/PENUP
FORWARD 10
PENDOWN

FORWARD 10
RIGHT 90
FORWARD