Algorithm design and problem-solving – 2020 IGCSE 0478

1. Nov/2020/Paper_21/No.1(a),(b)

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A car insurance system allows customers to check how much it would cost to buy insurance for a car for twelve months. The starting price of the car insurance is $500. The actual price a customer pays for the car insurance changes depending upon this information:

- engine size of car
- value of car
- where car is kept overnight
- number of kilometres (km) driven a year
- age of driver
- years insured without an insurance claim

Engine size in litres	Price change	Value in $1000	Price change	Kept over-night	Price change	1000 km driven a year	Price change	Age of driver	Price change
<=0.5	–5%	<0.5	–5%	Garage	–5%	<5	–5%	18–20	+100%
>0.5 to 1.0	0%	0.5 to 2	0%	Drive	0%	5 to 20	0%	21–25	+50%
>1.0 to 2.5	+5%	>2 to 10	+5%	Street	+5%	>20	+5%	26–30	+25%
>2.5	+10%	>10 to 20	+10%					31–70	0%
		>20	+15%					71–80	+10%
								>80	+20%

Table 1

Years without claim	Price change
1	–10%
2	–20%
3	–30%
4	–40%
5	–50%
6	–60%
>6	–70%

Table 2

The actual price is calculated by:

- finding the total of the percentage changes for the customer using Table 1
- applying this total percentage change to the starting price of the car insurance
- applying the years without claim discount percentage for the customer using Table 2.

Write and test a program or programs to calculate the price for a customer to insure a car.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – Calculate the price to insure a car.

Write a program to obtain the required information from a customer and calculate the price to insure the car. Display the price to insure the car. Display the total percentage change calculated from Table 1, and the years without claim discount percentage from Table 2 separately.

Task 2 – New customer discount.

Extend **Task 1** to include an additional discount of 10% off the price to any new customer who is aged between 26 and 70 inclusive, who also has 2 or more years without a claim. Display the amount of money this would save and the new price.

Task 3 – Adding an extra driver.

Customers can add one extra driver. The age of the extra driver may increase the price. The new price is calculated by applying the percentage price change for the age of the extra driver from Table 1 to the price. Extend **Task 2** to calculate and display the new price including an extra driver if required.

All variables, constants and other identifiers must have meaningful names.

(a) Identify **one** array that you could have used for **Task 1**. Give the data type and state the use of the array.

Array ....

Data type ....

Use ....

....

....
[3]

(b) Data entered by a customer includes the age of the driver in **Task 1**.
Identify **two** different validation checks you could use for the age of the driver.
Justify your choice for each one.

Validation check 1 ....

....

Justification ....

....

....

....

Validation check 2 ....

....

Justification ....

....

....

....
[4]

2. Nov/2020/Paper_21/No.2

Describe the purpose of variables and constants. Use an example of each in your answer.

....

....

....

....

....

....

....

....

.... [4]

3. Nov/2020/Paper_21/No.3

This pseudocode algorithm calculates the weight and number of bags in a load of firewood. The weight in kilograms of each bag is input. The algorithm finishes when either 50 bags have been weighed, or as soon as the total weight exceeds 1000 kilograms. Only then are the total weight and the number of bags in the load output.

```
01  TotalWeight ← 1000
02  BagCount ← 0
03  MaxBag ← 50
04  MaxWeight ← 1000
05  REPEAT
06      OUTPUT "Please Enter weight of bag"
07      INPUT Weight
08      TotalWeight ← TotalWeight + Weight
09      BagCount ← BagCount + 1
10      OUTPUT "Number of bags in the load is ", BagCount
11  UNTIL TotalWeight > MaxWeight AND BagCount >= MaxBag
12  OUTPUT "Total weight of the load is ", MaxWeight
```

(a) Give the line number(s) from the algorithm of:

an assignment statement ....

a loop ....

a counting statement ....

a totalling statement .... [4]

(b) Give the line numbers of the **four** errors in this pseudocode. Suggest a correction for each error.

Error 1 line number ....

Correction ....

....

Error 2 line number ....

Correction ....

....

Error 3 line number ....

Correction ....

....

Error 4 line number ....

Correction ....

.... [4]

(c) Explain how you could extend the algorithm to calculate and display the average weight of a bag of firewood in the load.

....

....

....

....

....

....

....

....

....

....

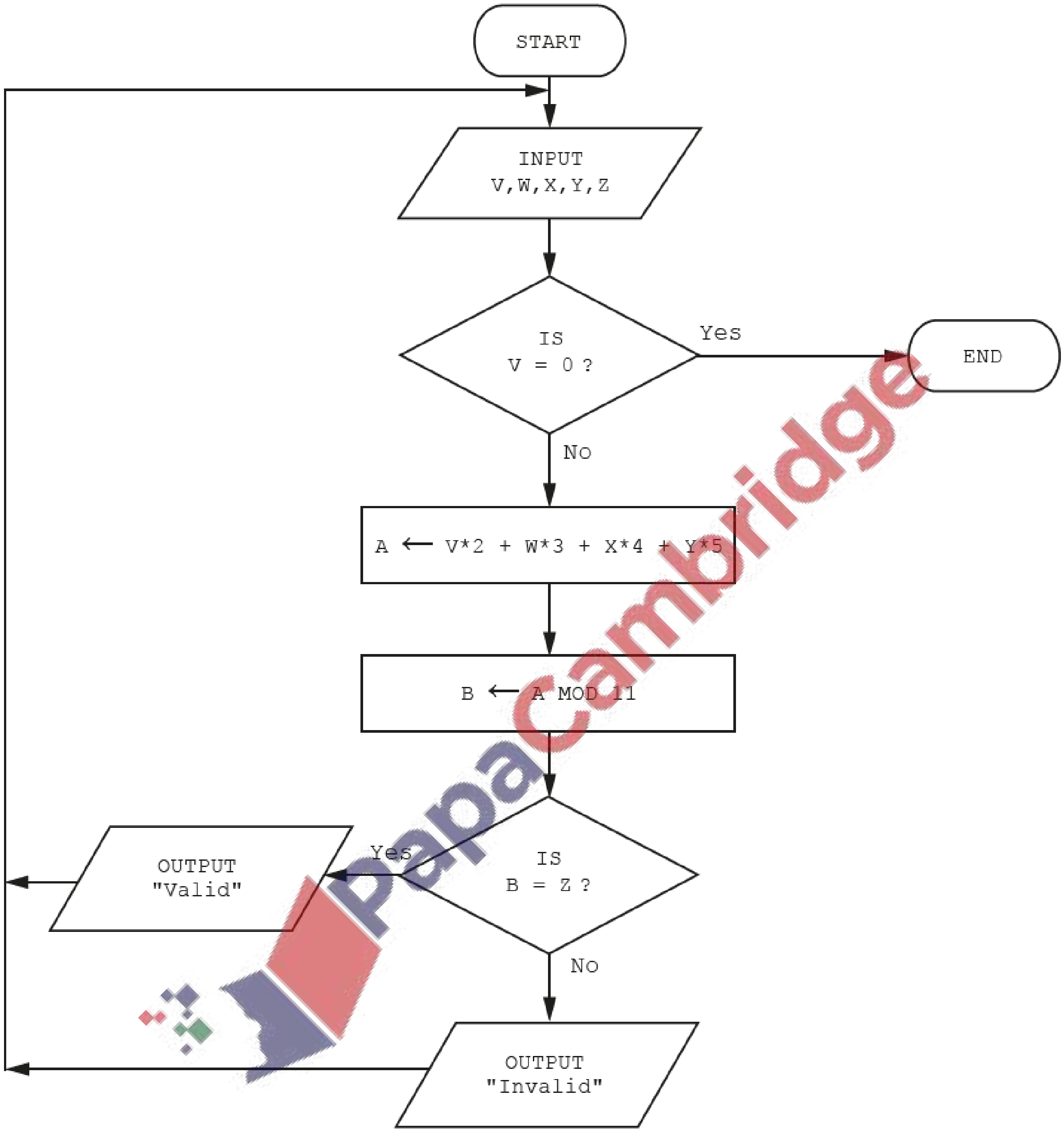
....

.... [4]

4. Nov/2020/Paper_21/No.4

This flowchart inputs five numbers and performs a calculation.

The predefined function MOD finds the remainder from integer division for example `R ← 25 MOD 11` gives `R` a value of `3`



(a) Complete the trace table for this set of input data:
5, 4, 6, 2, 1, 9, 3, 2, 1, 6, 7, 6, 1, 5, 1, 0, 0, 0, 0, 0

V	W	X	Y	Z	A	B	OUTPUT

[4]

(b) Describe the purpose of this flowchart.

....

....

....

.... [2]

5. Nov/2020/Paper_22/No.1(a),(b),(d)

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

An online computer shop sells customised personal computers. Every computer sold includes a basic set of components costing $200 and additional items can be added from the table:

Category	Item code	Description	Price ($)
Case	A1	Compact	75.00
Case	A2	Tower	150.00
RAM	B1	8 GB	79.99
RAM	B2	16 GB	149.99
RAM	B3	32 GB	299.99
Main Hard Disk Drive	C1	1 TB HDD	49.99
Main Hard Disk Drive	C2	2 TB HDD	89.99
Main Hard Disk Drive	C3	4 TB HDD	129.99
Solid State Drive	D1	240 GB SSD	59.99
Solid State Drive	D2	480 GB SSD	119.99
Second Hard Disk Drive	E1	1 TB HDD	49.99
Second Hard Disk Drive	E2	2 TB HDD	89.99
Second Hard Disk Drive	E3	4 TB HDD	129.99
Optical Drive	F1	DVD/Blu-Ray Player	50.00
Optical Drive	F2	DVD/Blu-Ray Re-writer	100.00
Operating System	G1	Standard Version	100.00
Operating System	G2	Professional Version	175.00

As well as the basic set of components every computer must include one case, one RAM and one Main Hard Disk Drive from the table.
A computer is supplied with or without an Operating System.

Write and test a program or programs for the online computer shop.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All arrays, variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – Setting up the system and ordering the main items.
Write a program to:

- use arrays to store the item code, description and price
- allow a customer to choose one case, one RAM and one Main Hard Disk Drive
- calculate the price of the computer using the cost of the chosen items and the basic set of components
- store and output the chosen items and the price of the computer.

Task 2 – Ordering additional items.
Extend TASK 1 to:

- allow a customer to choose whether to purchase any items from the other categories – if so, which item(s)
- update the price of the computer
- store and output the additional items and the new price of the computer.

Task 3 – Offering discounts.
Extend TASK 2 to:

- apply a 5% discount to the price of the computer if the customer has bought only one additional item
- apply a 10% discount to the price of the computer if the customer has bought two or more additional items
- output the amount of money saved and the new price of the computer after the discount.

All variables, constants and other identifiers must have meaningful names.

(a) **(i)** Identify **one** array you could have used for **Task 1** and state its purpose.

Array ....

Purpose ....

....

.... [2]

(ii) Identify **one** variable you could have used for **Task 2** and state its purpose.

Variable ....

Purpose ....

....

.... [2]

(iii) Identify **one** constant you could have used for **Task 3** and state its purpose.

Constant ....

Purpose ....

....

.... [2]

(b) Explain the benefits of storing Price as a real data type.

....

....

....

....

....

....

.... [2]

(d) Explain how your program completes **Task 3**. Any programming statements used in your answer must be fully explained.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [4]

6. Nov/2020/Paper_22/No.2

Tick (✓) **one** box in each row to identify if the statement about subroutines is **true** or **false**.

Statement	true (✓)	false (✓)
A subroutine is called from within a program.		
A subroutine is **not** a complete program.		
A subroutine is a self-contained piece of code.		
A subroutine must return a value to the code from which it was called.		

[2]

7. Nov/2020/Paper_22/No.3

This pseudocode algorithm is used as a validation check.

```
PRINT "Input a number from 1 to 5000"
REPEAT
  INPUT Number
  IF Number < 1 OR Number > 5000
    THEN
      PRINT "Invalid number, please try again"
  ENDIF
UNTIL Number >= 1 AND Number <= 5000
PRINT Number, " is within the correct range"
```

Identify **three** different types of test data. For each type, give an example of the test data you would use to test this algorithm and state a reason for your choice of test.

Type of test data 1 ....

Test data ....

Reason ....

....

Type of test data 2 ....

Test data ....

Reason ....

....

Type of test data 3 ....

Test data ....

Reason ....

....

[6]

8. Nov/2020/Paper_22/No.5

This pseudocode represents an algorithm.

```
REPEAT
  Flag ← 0
  FOR Count ← 0 to 3
    IF Num[Count] < Num[Count + 1]
      THEN
        Store ← Num[Count]
        Num[Count] ← Num[Count + 1]
        Num[Count + 1] ← Store
        Flag ← 1
    ENDIF
  NEXT Count
UNTIL Flag = 0
```

(a) The contents of the array at the start of the algorithm are:

Num[0]	**Num[1]**	**Num[2]**	**Num[3]**	**Num[4]**
45	56	30	12	15

Complete the trace table for the algorithm using the data given in the array.

Flag	Count	Num[0]	Num[1]	Num[2]	Num[3]	Num[4]	Store
		45	56	30	12	15	

[5]

(b) Describe the purpose of the algorithm.

....

....

....

.... [2]

9. Nov/2020/Paper_22/No.6

Draw a flowchart symbol to represent each of the following:

Input/Output	Decision

[2]

10. Nov/2020/Paper_23/No.1(a),(b)

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A baguette ordering service allows customers to order filled baguettes. There are two sizes of baguette: 30 cm and 15 cm. Baguettes are available as white, brown or seeded bread. Baguettes have one filling and can have up to three salad items added.

Filling and salad choices are:

Filling	Salad
Beef	Lettuce
Chicken	Tomato
Cheese	Sweetcorn
Egg	Cucumber
Tuna	Peppers
Turkey	

Customers choose their baguette options. They then confirm their order, alter their choices or choose not to proceed.

Each day the ordering service calculates the number of each:

- size of baguette sold
- type of bread (white, brown or seeded) sold
- filling sold.

Write and test a program or programs for the baguette ordering service.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – Ordering.

Customers are given choices on size, type of bread, filling and salad items as shown. Only valid choices can be accepted. The customer is asked to confirm their order, alter their choices or choose not to proceed. If the customer confirms their order they are given a unique order number. Display the baguette ordered and the order number.

Task 2 – Recording the choices.

Extend TASK 1 to record totals for the size, types of bread and fillings sold that day and calculate the total number of baguettes sold that day.

Task 3 – Finding the most and least popular baguette fillings.

Using your results from TASK 2, display the most popular and least popular fillings as a percentage of the total number of baguettes sold that day.

All variables, constants and other identifiers must have meaningful names.

(a) Identify **one** constant and identify **one** variable that you could have used for **Task 1**. Give the value that would be assigned to the constant. Give the data type for the variable. State what each one could be used for.

Constant ....

Value ....

Use ....

....

Variable ....

Data type ....

Use ....

....

[6]

(b) **(i)** Write an algorithm to allow a customer to choose the filling and salad items for their baguette (part of **Task 1**), using **either** pseudocode, programming statements **or** a flowchart. Your algorithm must only include this part of **Task 1**.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [5]

(ii) Explain how your algorithm in **part (b)(i)** ensured that only valid choices were accepted for the filling.

....

....

....

....

....

....

.... [3]

11. Nov/2020/Paper_23/No.2

An algorithm has been written in pseudocode to check the temperature readings taken from a freezer are within the range –18 degrees to –25 degrees inclusive.

The algorithm counts the number of times that the temperature reading is below –25 degrees and the number of times that the temperature reading is above –18 degrees.

An engineer is called if there are more than 10 temperature readings below –25 degrees.

An alarm sounds if there are more than 5 temperature readings above –18 degrees.

```
01  TooHot ← 0
02  TooCold ← 1000
03  REPEAT
04      OUTPUT "Please enter temperature"
05      INPUT Temperature
06      IF Temperature < -25
07        THEN
08          TooCold ← TooCold - 1
09      ENDIF
10      IF Temperature > -18
11        THEN
12          TooHot ← TooHot + 1
13      ENDIF
14  UNTIL TooHot > 5 OR TooCold > 10
15  IF TooHot < 5
16    THEN
17      INPUT "Alarm!!"
18  ENDIF
19  IF TooCold > 10
20    THEN
21      OUTPUT "Call the Engineer"
22  ENDIF
```

(a) Give the line number(s) from the algorithm of:

an assignment statement ....

a loop ....

a counting statement ....

a selection statement ....

[4]

(b) Give line numbers where the **four** errors are to be found in the pseudocode. Suggest a correction for each error.

Error 1 line number ....

Correction ....

....

Error 2 line number ....

Correction ....

....

Error 3 line number ....

Correction ....

....

Error 4 line number ....

Correction ....

....

[4]

(c) Explain how you could extend the algorithm to count the number of times the temperature readings are within the range –18 degrees to –25 degrees inclusive.

....

....

....

....

....

....

....

....

....

.... [4]

12. Nov/2020/Paper_23/No.3

Four programming concepts and five descriptions are shown.

Draw a line to connect each **Programming concept** to its correct **Description**. Not all Descriptions will be connected to a Programming concept.

Programming concept

- Validation
- Verification
- Procedure
- Function

Description

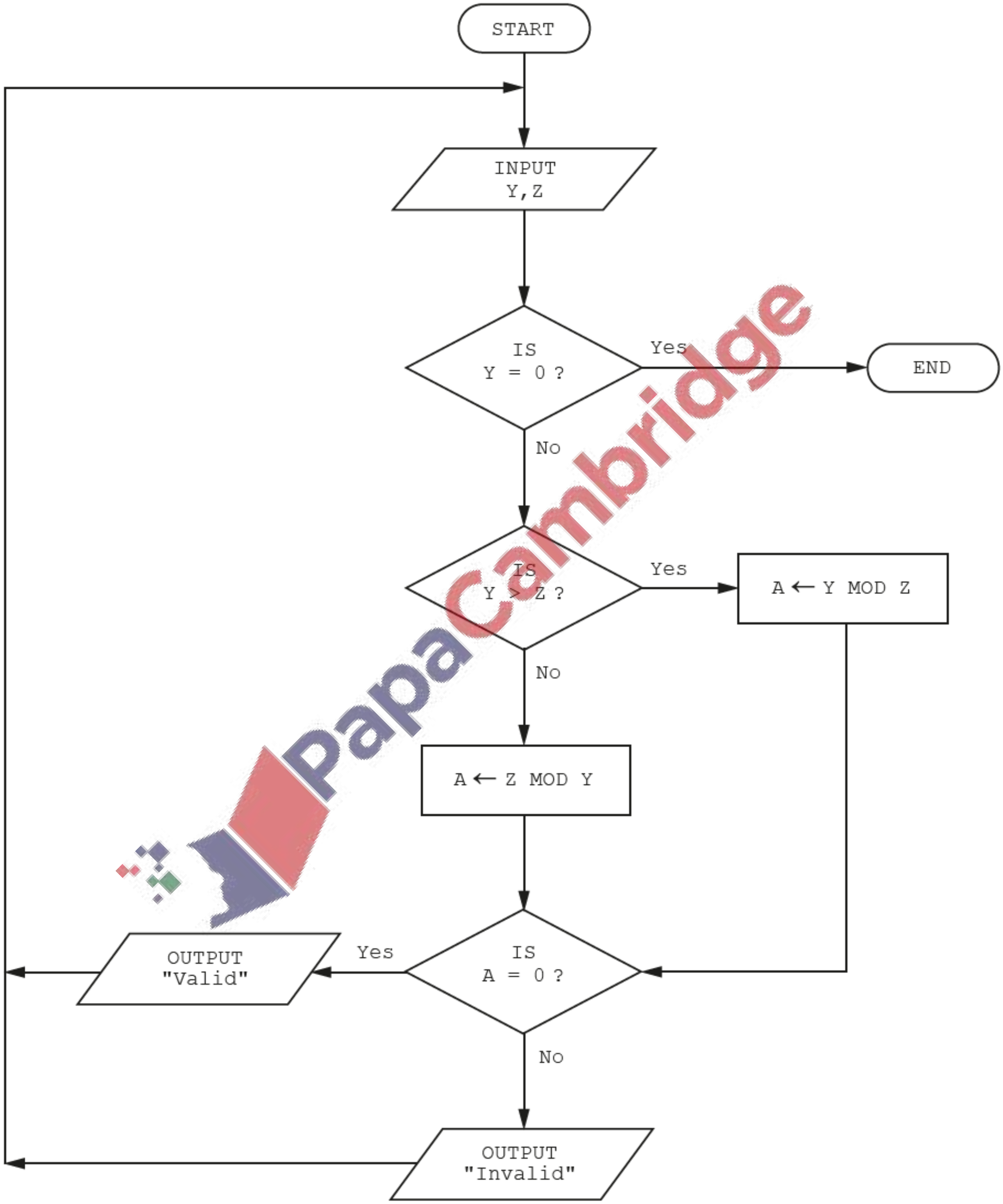
- A subroutine that does not have to return a value
- An automatic check to ensure that data input is reasonable and sensible
- A subroutine that always returns a value
- An overview of a program or subroutine
- A check to ensure that data input matches the original

[4]

13. Nov/2020/Paper_23/No.4

This flowchart represents an algorithm that allows the input of two numbers and performs a calculation.

The predefined function MOD finds the remainder from integer division for example X ← 8 MOD 5 gives X a value of 3.



(a) Complete a trace table for this set of input data:
11, 4, 6, 2, 3, 9, 3, 2, 2, 6, 0, 0, 1, 1

Y	Z	A	OUTPUT

[4]

(b) Explain the purpose of this algorithm.

....

....

....

.... [2]

14. June/2020/Paper_21/No.1(a),(b),(d)

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A shop sells a range of mobile devices, SIM cards and accessories as shown in the table:

Category	Item code	Description	Price ($)
Phone	BPCM	Compact	29.99
Phone	BPSH	Clam Shell	49.99
Phone	RPSS	RoboPhone – 5-inch screen and 64 GB memory	199.99
Phone	RPLL	RoboPhone – 6-inch screen and 256 GB memory	499.99
Phone	YPLS	Y-Phone Standard – 6-inch screen and 64 GB memory	549.99
Phone	YPLL	Y-Phone Deluxe – 6-inch screen and 256 GB memory	649.99
Tablet	RTMS	RoboTab – 8-inch screen and 64 GB memory	149.99
Tablet	RTLM	RoboTab – 10-inch screen and 128 GB memory	299.99
Tablet	YTLM	Y-Tab Standard – 10-inch screen and 128 GB memory	499.99
Tablet	YTLL	Y-Tab Deluxe – 10-inch screen and 256 GB memory	599.99
SIM card	SMNO	SIM Free (no SIM card purchased)	0.00
SIM card	SMPG	Pay As You Go (SIM card purchased)	9.99
Case	CSST	Standard	0.00
Case	CSLX	Luxury	50.00
Charger	CGCR	Car	19.99
Charger	CGHM	Home	15.99

Write and test a program or programs for this shop.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All arrays, variables, constants and other identifiers must have meaningful names.

You will need to complete these **three tasks**. Each task must be fully tested.

Task 1 – Setting up the system.
Write a program to:

- use appropriate data structures to store the item code, description and price information for the mobile devices, SIM cards and accessories
- allow the customer to choose a specific phone or tablet
- allow phone customers to choose whether the phone will be SIM Free or Pay As You Go
- allow the customer to choose a standard or luxury case
- allow the customer to choose the chargers required (none, one or both may be purchased)
- calculate the total price of this transaction
- output a list of the items purchased and the total price.

Task 2 – Allow a customer to order multiple mobile devices.
Extend **Task 1** to:

- offer the customer the opportunity to purchase an additional mobile device
- if required, perform bulleted steps 2 to 7 of **Task 1** for each additional mobile device and calculate a running total for the customer
- once no further devices are required, output the total the customer will need to pay.

Task 3 – Offering discounts.
Extend the program to allow a discount of 10% off the price of every additional phone or tablet purchased.
Output the new total the customer will need to pay and the amount of money saved.

All variables, constants and other identifiers must have meaningful names.

(a) **(i)** Identify **two** arrays you could have used for **Task 1** and, in each case, state its purpose.

Array 1 ....

Purpose ....

....

....

Array 2 ....

Purpose ....

....

.... [4]

(ii) Identify **two** variables you could have used for **Task 1** and, in each case, state its purpose.

Variable 1 ....

Purpose ....

....

....

Variable 2 ....

Purpose ....

....

.... [4]

(b) Explain why the item code data could **not** be stored as a real data type and identify the most suitable data type for the item code data.

....

....

....

....

....

.... [2]

(d) Explain how your program completes **Task 3**. Any programming statements used in your answer must be fully explained.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [5]

15. June/2020/Paper_21/No.2

Tick (✓) **one** box in each row to identify if the statement about structure diagrams is true or false.

Statement	True (✓)	False (✓)
A structure diagram is a piece of code that is available throughout the structure of a program.		
A structure diagram shows the hierarchy of a system.		
A structure diagram is another name for an array.		
A structure diagram shows the relationship between different components of a system.		

[2]

16. June/2020/Paper_21/No.3

Programs can perform validation and verification checks when data is entered.

(a) Give the names of **two** different validation checks and state the purpose of each one.

Check 1 ....

Purpose ....

....

....

Check 2 ....

Purpose ....

....

....

[4]

(b) Give the name of **one** verification check.

.... [1]

(c) Describe the difference between validation and verification.

....

....

....

.... [2]

17. June/2020/Paper_21/No.4(a),(c)

The pseudocode algorithm shown should allow numbers to be entered and should allow 50 numbers to be stored in an array.

```
Count ← 0
REPEAT
   INPUT Values[Count]
   Count ← Count + 1
UNTIL Count = 0
```

(a) Explain why the algorithm will never end.

....

....

....

....

.... [2]

(c) Describe how you could change your pseudocode in **part (b)** so that it prevents numbers below 100 and above 200 from being stored in the array `Values[ ]`

....

....

....

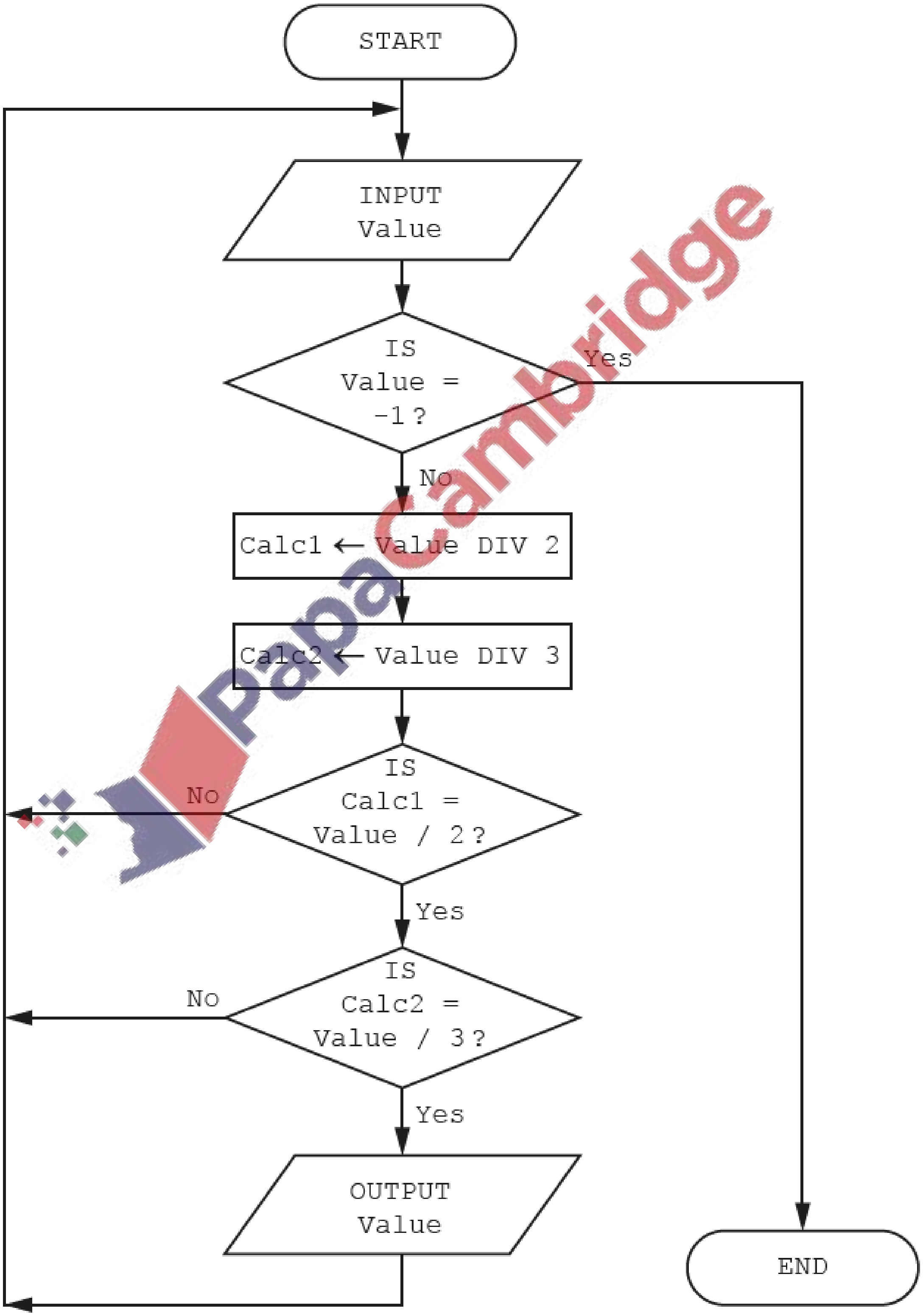
.... [2]

18. June/2020/Paper_21/No.5

The flowchart represents an algorithm.

The predefined function `DIV` gives the value of the result of integer division, for example, `y ← 9 DIV 4` gives `y` a value of `2`

An input value of –1 ends the algorithm.



(a) Complete the trace table for the input data:

50, 33, 18, 15, 30, –1, 45, 12, 90, 6

`Value`	`Calc1`	`Calc2`	**OUTPUT**

[4]

(b) Describe the purpose of the algorithm.

....

....

.... [2]

19. June/2020/Paper_22/No.1(a),(b),(c)

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A car park payment system allows customers to select the number of hours to leave their car in the car park. The customer will get a discount if they enter their frequent parking number correctly. The system calculates and displays the amount the customer must pay. The price of parking, the number of hours the customer can enter, and any discount depend upon the day of the week and the arrival time. The number of hours entered is a whole number. The price per hour is calculated using the price in force at the arrival time. No parking is allowed between Midnight and 08:00.

Day of the week	**Arrival time**			
	From 08:00 to 15:59		**From 16:00 to Midnight**	
	Max stay in hours	**Price per hour**	**Hours**	**Price**
Sunday	8	2.00	Up to Midnight	2.00
Monday	2	10.00	Up to Midnight	2.00
Tuesday	2	10.00	Up to Midnight	2.00
Wednesday	2	10.00	Up to Midnight	2.00
Thursday	2	10.00	Up to Midnight	2.00
Friday	2	10.00	Up to Midnight	2.00
Saturday	4	3.00	Up to Midnight	2.00

A frequent parking number can be entered for discounted parking. This number consists of 4 digits and a check digit that is calculated using a modulo 11 check digit calculation. A discount of 50% is available for arrival times from 16:00 to Midnight; the discount is 10% at all other arrival times.

Write and test a program or programs to simulate the car park payment system.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – Calculating the price to park.

A customer inputs the day, the hour of arrival excluding minutes (for example 15:45 would be 15), the number of hours to leave their car, and a frequent parking number if available. If the frequent parking number has an incorrect check digit, then no discount can be applied. The price to park, based on the day, the hour of arrival, the number of hours of parking required and any discount available, is calculated and displayed.

Task 2 – Keeping a total of the payments.

Extend **Task 1** to keep a daily total of payments made for parking. The daily total is zeroed at the start of the day. For the simulation, each customer inputs the amount paid, this must be greater than or equal to the amount displayed. There is no change given so the amount input may exceed the amount displayed. Each customer payment is added to the daily total, and this total is displayed at the end of the day.

Task 3 – Making payments fairer.

Customers have complained that sometimes they are being charged too much if they arrive before 16:00 and depart after 16:00. Extend **Task 1** to calculate the price before 16:00, then add the evening charge. For example, a customer arriving at 14:45 on a Sunday and parking for five hours was previously charged 10.00 and would now be charged 6.00

All variables, constants and other identifiers must have meaningful names.

(a) Identify **two** variables that you could have used for **Task 1**. Give the data type and state the use of each variable.

Variable 1 ....

Data type ....

Use ....

....

Variable 2 ....

Data type ....

Use ....

.... [4]

(b) Data input by a customer for **Task 1** includes the day and the hour of arrival.

Identify **one** suitable validation check for each input and justify your choice. Your validation checks must be different.

Day of arrival – validation check ....

....

Justification ....

....

....

....

Hour of arrival – validation check ....

....

Justification ....

....

....

.... [4]

(c) Explain how your program for **Task 1** ensured that the frequent parking number entered by the customer had a valid check digit.

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [4]

20. June/2020/Paper_22/No.2

Most programming languages include basic data types. Ahmad is describing the basic data types he has used.

State the data type that Ahmad is describing in each sentence.

Choose the data type from this list of programming terms.

Array **Boolean** **Char** **Constant** **Function** **Integer**

Iteration **Procedure** **Real** **String** **Variable**

A number with a fractional part that can be positive or negative and used in calculations

Data type ....

A whole number that can be positive, negative or zero and used in calculations

Data type ....

A single number, symbol or letter

Data type ....

A sequence of characters

Data type ....

A data type with two values, True or False

Data type ....

[5]

21. June/2020/Paper_22/No.3

(a) An algorithm has been written in pseudocode to input the names and marks of 35 students. The algorithm stores the names and marks in two arrays `Name[ ]` and `Mark[ ]`. The highest mark awarded is found and the number of students with that mark is counted. Both of these values are output.

```
01  HighestMark ← 100
02  HighestMarkStudents ← 0
03  FOR Count ← 1 TO 35
04      OUTPUT "Please enter student name"
05      INPUT Name[Count]
06      OUTPUT "Please enter student mark"
07      INPUT Mark[Counter]
08      IF Mark[Count] = HighestMark
09        THEN
10          HighestMarkStudents ← HighestMarkStudents – 1
11      ENDIF
12      IF Mark[Count] > HighestMark
13        THEN
14          Mark[Count] ← HighestMark
15          HighestMarkStudents ← 1
16      ENDIF
17  NEXT Count
18  OUTPUT "There are ", HighestMarkStudents," with the highest mark of ",
      HighestMark
```

Give line numbers where the **four** errors are to be found in the pseudocode. Suggest a correction for each error.

Error 1 line number ....

Correction ....

....

Error 2 line number ....

Correction ....

....

Error 3 line number ....

Correction ....

....

Error 4 line number ....

Correction ....

....

[4]

(b) Explain how you could extend the algorithm to also find the lowest mark awarded, count the number of students with that mark, and output both these values.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

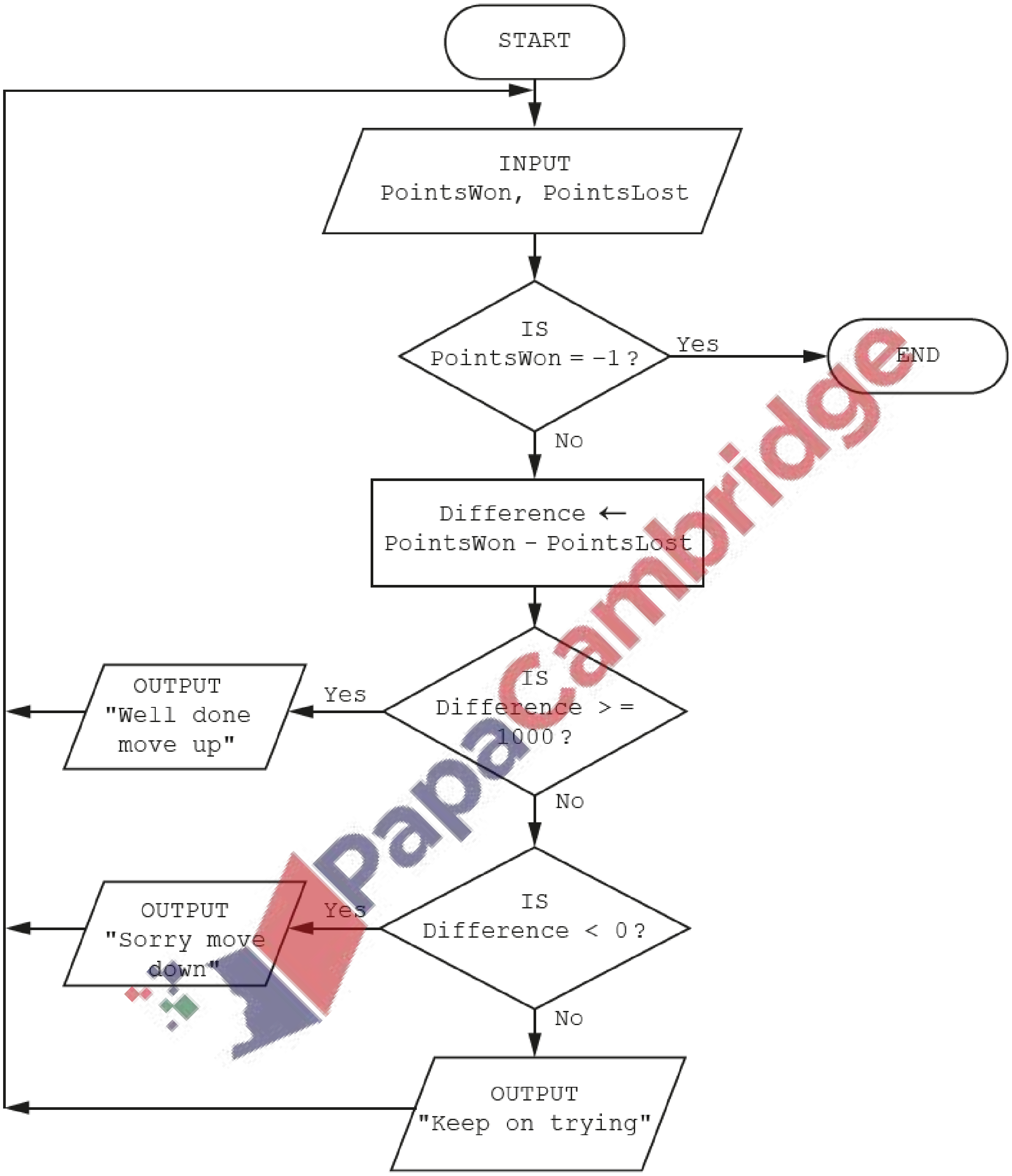
....

....

.... [6]

22. June/2020/Paper_22/No.4

This flowchart inputs the points won and the points lost when playing a game. The difference between the points won and lost is calculated and depending on the result the player can: move up to the next level, stay at the same level, or move down to the previous level. The flowchart finishes when the input for points won is –1.



(a) Complete a trace table for this set of input data:
5000, 4474, 6055, 2000, 7900, 9800, 3000, 2150, –1, 6700, 7615

`PointsWon`	`PointsLost`	`Difference`	OUTPUT

[3]

(b) The flowchart needs to be changed. When the difference is more than 5000 the output message is 'Fantastic leap up two levels'.

Describe the changes that will need to be made to the flowchart.

....

....

....

....

....

....

....

.... [3]

23. June/2020/Paper_22/No.5

Arrays are data structures used in programming. Explain what is meant by the terms dimension and index in an array. Use examples of arrays in your explanations.

Dimension ....

....

....

....

....

Index ....

....

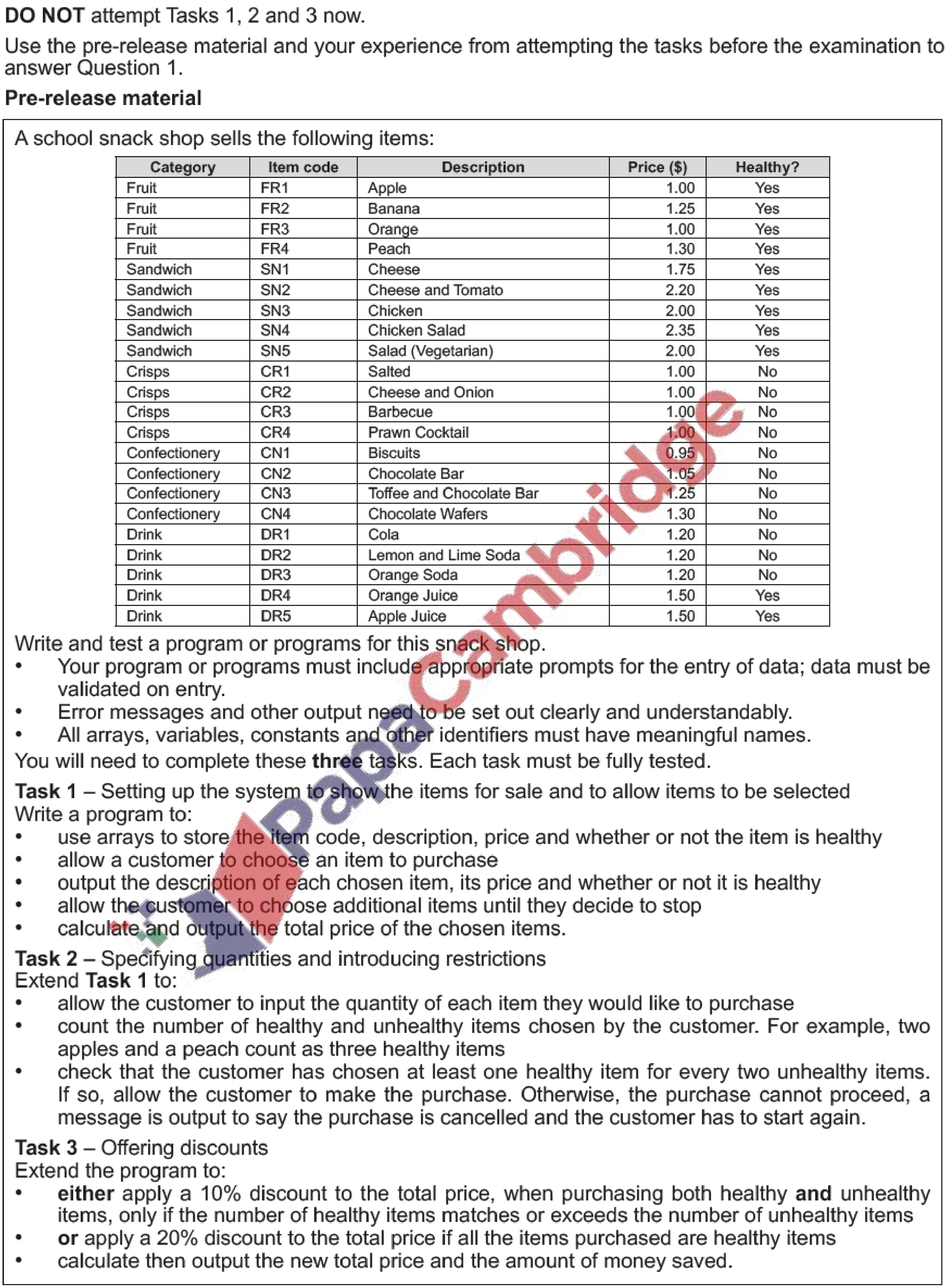
....

....

....

[3]

24. June/2020/Paper_23/No.1(a),(b),(d),(e)



DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A school snack shop sells the following items:

Category	Item code	Description	Price ($)	Healthy?
Fruit	FR1	Apple	1.00	Yes
Fruit	FR2	Banana	1.25	Yes
Fruit	FR3	Orange	1.00	Yes
Fruit	FR4	Peach	1.30	Yes
Sandwich	SN1	Cheese	1.75	Yes
Sandwich	SN2	Cheese and Tomato	2.20	Yes
Sandwich	SN3	Chicken	2.00	Yes
Sandwich	SN4	Chicken Salad	2.35	Yes
Sandwich	SN5	Salad (Vegetarian)	2.00	Yes
Crisps	CR1	Salted	1.00	No
Crisps	CR2	Cheese and Onion	1.00	No
Crisps	CR3	Barbecue	1.00	No
Crisps	CR4	Prawn Cocktail	1.00	No
Confectionery	CN1	Biscuits	0.95	No
Confectionery	CN2	Chocolate Bar	1.05	No
Confectionery	CN3	Toffee and Chocolate Bar	1.25	No
Confectionery	CN4	Chocolate Wafers	1.30	No
Drink	DR1	Cola	1.20	No
Drink	DR2	Lemon and Lime Soda	1.20	No
Drink	DR3	Orange Soda	1.20	No
Drink	DR4	Orange Juice	1.50	Yes
Drink	DR5	Apple Juice	1.50	Yes

Write and test a program or programs for this snack shop.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All arrays, variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

Task 1 – Setting up the system to show the items for sale and to allow items to be selected
Write a program to:

- use arrays to store the item code, description, price and whether or not the item is healthy
- allow a customer to choose an item to purchase
- output the description of each chosen item, its price and whether or not it is healthy
- allow the customer to choose additional items until they decide to stop
- calculate and output the total price of the chosen items.

Task 2 – Specifying quantities and introducing restrictions
Extend **Task 1** to:

- allow the customer to input the quantity of each item they would like to purchase
- count the number of healthy and unhealthy items chosen by the customer. For example, two apples and a peach count as three healthy items
- check that the customer has chosen at least one healthy item for every two unhealthy items. If so, allow the customer to make the purchase. Otherwise, the purchase cannot proceed, a message is output to say the purchase is cancelled and the customer has to start again.

Task 3 – Offering discounts
Extend the program to:

- **either** apply a 10% discount to the total price, when purchasing both healthy **and** unhealthy items, only if the number of healthy items matches or exceeds the number of unhealthy items
- **or** apply a 20% discount to the total price if all the items purchased are healthy items
- calculate then output the new total price and the amount of money saved.

All variables, constants and other identifiers must have meaningful names.

(a) (i) Identify **one** array you could have used for **Task 1** and state its purpose.

Array ....

Purpose ....

....

.... [2]

(ii) Identify **one** variable you could have used for **Task 2** and state its purpose.

Variable ....

Purpose ....

....

.... [2]

(iii) Identify **one** constant you could have used for **Task 3** and state its purpose.

Constant ....

Purpose ....

....

.... [2]

(b) Explain why a student would be advised **not** to store the Healthy? data as a real data type and identify the most suitable data type for the Healthy? data.

....

....

....

....

....

.... [2]

(d) Explain how your program completes **Task 3**. Assume **Task 1** and **Task 2** have been completed. Any programming statements used in your answer must be fully explained.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [4]

(e) Describe how you could alter your program to restrict the maximum number of items purchased by a customer to **six**.

....

....

....

....

....

....

....

....

....

....

....

.... [2]

25. June/2020/Paper_23/No.2

Draw a line to connect each **Description** to the most appropriate **Pseudocode example**.

Description	**Pseudocode example**
A loop that will iterate at least once	`CASE … OF … OTHERWISE … ENDCASE`
A loop that will not be executed on the first test if the condition is false	`Number ← Number + 1`
A conditional statement	`WHILE … DO … ENDWHILE`
Totalling	`Sum ← Sum + NewValue`
Counting	`REPEAT … UNTIL`

[4]

26. June/2020/Paper_23/No.3

This section of pseudocode is to be used as a validation check that will continue until a number between 0 and 499 inclusive is entered.

```
1       PRINT "Input a number from 0 to 499 inclusive"
2       FOR Number ← 1 TO 10
3         INPUT Number
4         IF Number < 0 AND Number > 499
5           THEN
6             PRINT "Invalid number, please try again"
7         ENDIF
8       UNTIL Number = 0 OR Number = 499
9       PRINT Number, " is within the correct range"
```

There are **three** lines in this pseudocode that contain errors. In each case, state the line number to identify the incorrect line and write out the corrected line in full.

Error 1 line number ....

Correction ....

....

Error 2 line number ....

Correction ....

....

Error 3 line number ....

Correction ....

....

[6]

27. June/2020/Paper_23/No.4

A code must take the form LL9 9LL where L is a letter and 9 is a digit.

(a) A presence check has already been used to ensure data has been entered. Name **two** other types of validation check that can be used to test the code is valid.

Check 1 ....

Check 2 ....
[2]

(b) Give **one** example of invalid test data for each of the validation checks you have named in **part (a)** and in each case, give a reason why it fails the check. Each example of test data must be different.

Check 1 Invalid Test Data ....

....

Reason ....

....

....

....

Check 2 Invalid Test Data ....

....

Reason ....

....

....

....
[4]

28. June/2020/Paper_23/No.5

This algorithm finds prime numbers.

The pre-defined function `DIV` gives the value of the result of integer division, for example, `y ← 9 DIV 4` gives `y` a value of `2`

```
Flag ← False
INPUT Number
WHILE Number <> 0
  Divisor ← 2
  WHILE Divisor <= Number / 2
    Value ← Number DIV Divisor
    IF Number / Divisor = Value
      THEN
        Flag ← True
    ENDIF
    Divisor ← Divisor + 1
  ENDWHILE
  IF Flag = False
    THEN
      OUTPUT Number, " is prime"
  ENDIF
INPUT Number
Flag ← False
ENDWHILE
```

Complete the trace table for the algorithm using the input data:

5, 6, 8, 0, 11, 13

Flag	Number	Divisor	Value	OUTPUT

[5]

29. June/2020/Paper_23/No.6

Draw the flowchart symbol for **Decision** and the flowchart symbol for **Process**.

Decision	Process

[2]

30. March/2020/Paper_22/No.1(a),(b),(c),(e)

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A car buying service helps customers calculate the price of buying a new car from Snazzy Autos. There are three models of car available: hatchback, saloon and estate. A car can be bought with optional extras.

Model	Price
Hatchback	Rs 5.35 lakh
Saloon	Rs 4.95 lakh
Estate	Rs 6.25 lakh

Optional extra	Price
Set of luxury seats	Rs 45000
Satellite navigation	Rs 5500
Parking sensors	Rs 10000
Bluetooth connectivity	Rs 350
Sound system	Rs 1000

Snazzy Autos customers buying a new car can trade in an old car. Offers from Rs 10000 to Rs 1.00 lakh can be made for an old car. If an old car is not traded in, then a discount of 5% is taken off the price of the new car and any optional extras. In addition, repeat customers are offered a discount of 10% off any optional extras and 10% off the price of the new car before trade-in.
Customers can either pay for the new car in full or make monthly payments over several years.

Write and test a program or programs for the car buying service.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three tasks**. Each task must be fully tested.

Task 1 – Calculate the price

Customers choose the model of car and select any optional extras required. Only valid choices can be accepted for the model and the extras. The customer is asked if they are an existing customer and if they have an old car to trade in. If the customer trades in an old car the amount offered by Snazzy Autos is entered. The price of the new car including model and extras is displayed. Any discount or trade-in amounts are displayed and the total price to pay is calculated and displayed.

Task 2 – Calculate the payment(s)

Extend **Task 1** to display these payment methods:

- full amount paid now customer receives cashback of 1% on the total price
- equal monthly payments are made over four years with no extra charge
- the total price is increased by 5%; equal monthly payments are made over seven years.

For each payment method display: the total amount to pay, the amount of each payment, the number of payments and cashback where appropriate.

Task 3 – New full amount payment offer

Extend **Task 2** to offer customers who decide to pay the full amount now either the 1% cashback or the chosen optional extras free. Customers are shown the offer that is the best value first and the other as an alternative. Details of each offer must be displayed in full.

All variables, constants and other identifiers must have meaningful names.

(a) State the name of **one** constant that you could have used for **Task 1**. Give the value that would be assigned to the constant. State the use of this constant.

Constant name ....

Value ....

Use ....

....

[3]

(b) State the name of the variable that you used to store the amount offered for the trade-in of an old car in **Task 1**. Give the most appropriate data type for this variable. Explain how your program ensured that any data entered for the trade-in offer was valid.

Variable name ....

Data type ....

Validation ....

....

....

....

[4]

(c) Snazzy Autos has decided to add another model to the cars available in **Task 1**. The new model is a convertible, price Rs 6.75 lakh. Explain the changes required to your program for **Task 1**.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [4]

(e) Explain how your program completed **Task 3**. Assume that **Task 2** has been completed. Any programming statements used in your answer must be fully explained.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [4]

31. March/2020/Paper_22/No.2

(a) An algorithm has been written in pseudocode to input 50 numbers. Positive numbers are stored in the array `PosNum[ ]`. Negative numbers are stored in the array `NegNum[ ]`. Zeros are not included in the positive and negative counts.

```
Count ← 0
PosCount ← Count
NegCount ← Count
REPEAT
    INPUT Number
    IF Number > 0
      THEN
        PosCount ← PosCount + 1
        PosNum[PosCount] ← Number
      ELSE
        NegCount ← NegCount + 1
        NegNum[NegCount] ← Number
    ENDIF
    Count ← Count + 1
UNTIL Count >= 50
OUTPUT "There are ", PosCount," positive numbers"
OUTPUT "There are ", NegCount," negative numbers"
```

Describe the error in the pseudocode and write the correction for this error.

Error ....

....

....

....

Correction ....

....

....

....

[4]

(b) The algorithm needs to be changed so there is no limit to how many numbers can be input. When the number 9999 is input, the algorithm stops more numbers being input and outputs the results. The number 9999 is not to be stored nor counted as a positive number.

Explain how you would change the algorithm.

....

....

....

....

....

....

....

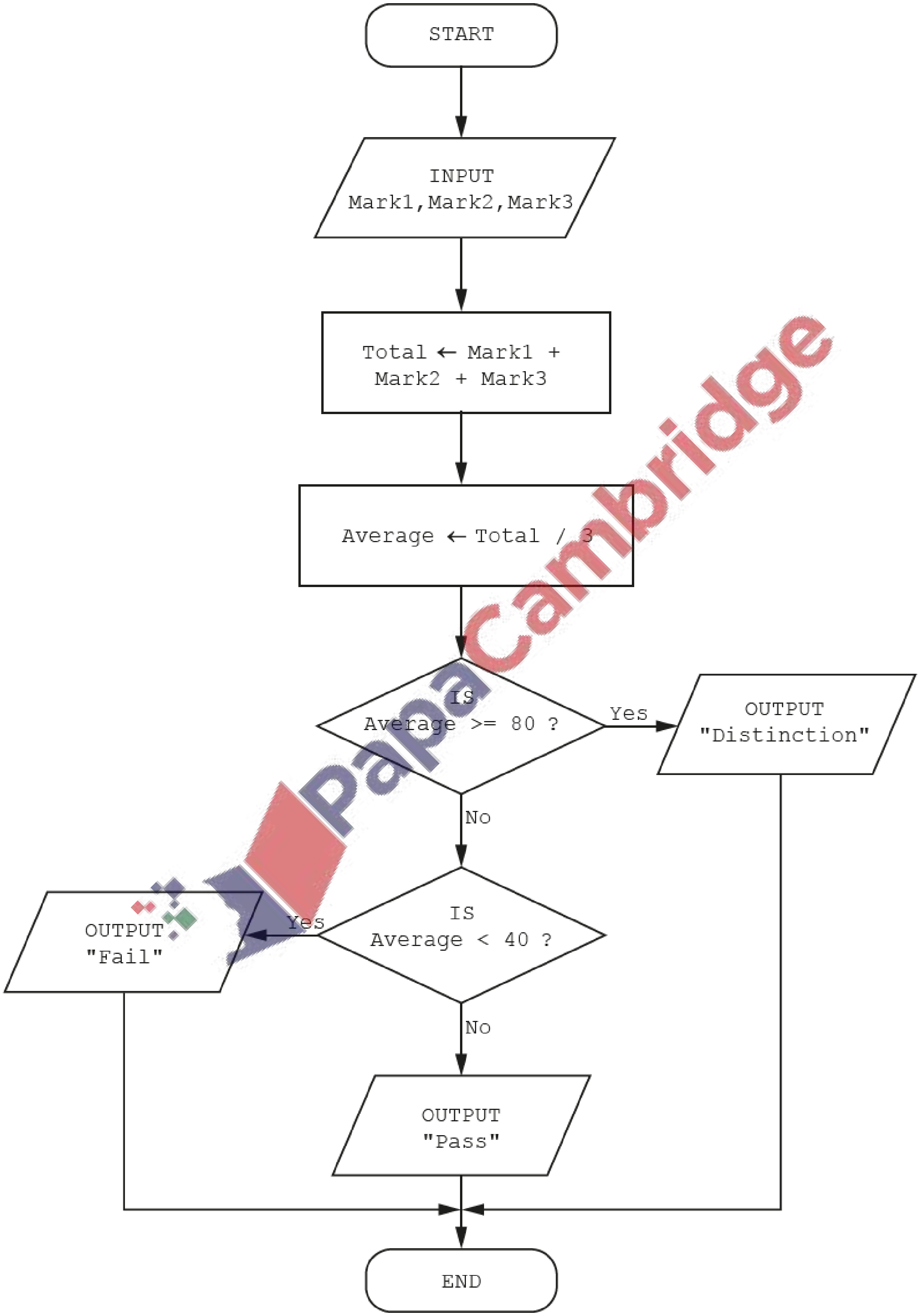
....

....

.... [4]

32. March/2020/Paper_22/No.3

This flowchart inputs student percentage marks for three examinations. If the average of these marks is 80% or over then a distinction grade is awarded. If the average of these marks is less than 40% then a fail grade is awarded. Otherwise a pass grade is awarded.



(a) Complete a trace table for each set of input data:

Set 1: 88, 74, 60

Mark1	Mark2	Mark3	Total	Average	OUTPUT

Set 2: 20, 33, 67

Mark1	Mark2	Mark3	Total	Average	OUTPUT

Set 3: 79, 91, 70

Mark1	Mark2	Mark3	Total	Average	OUTPUT

[5]

(b) It has been decided to include an extra grade of Merit when the average of the marks is 60% or more, and less than 80%. Describe the changes that will need to be made to the flowchart.

....

....

....

....

....

....

....

.... [3]

33. March/2020/Paper_22/No.4

Explain why validation and verification checks are needed when data is input.
Include an example of each type of check in your answer.

....

....

....

....

....

....

....

....

.... [4]