Chapter 16

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
Count ← 0 (5)
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 (2) should be assigned to "0"; the lowest number. High ← 0

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 (4) less than Low (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`.

~~counter~~ I would assign the variable Total ~~high~~ to "0" (5) in the begining, before ~~low~~ 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 ENDWHILE. 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 the 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]