Name: Index: Reg.no.: Class: Date:

JUNE 2023

1 (a) **Four** descriptions of stages in the program development life cycle are shown.

Draw **one** line to link each description to its most appropriate program development life cycle stage.

Not all program development life cycle stages will be used.

Program development life cycle description	Program development life cycle stage
develop an algorithm to solve the problem by using structure diagrams, flowcharts or pseudocode	analysis
detect and fix the errors in the program	coding
identify the problem and its requirements	design
write and implement the instructions to solve the problem	evaluation
	testing

[4]

(b) Identify **three** of the component parts after a problem has been decomposed.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

JUNE 2023 (2)

1 Tick (✓) **one** box to identify the first stage of the program development life cycle.

A Analysis ☐

B Coding ☐

C Design ☐

D Testing ☐

[1]

3 Identify **three** different ways that the design of a solution to a problem can be presented.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

NOVEMBER 2023

4 Circle the **three** words representing places where data may be stored.

array constant dimension input

output procedure variable

[3]

5 The first stage of the program development life cycle is analysis. Two of the tasks in analysis are abstraction and decomposition.

(a) Describe what is meant by abstraction.

.....

.....

.....

..... [2]

(b) Identify **three** of the component parts when a problem has been decomposed at the analysis stage.

1 .....

2 .....

3 .....

[3]

(c) Identify and describe **one** other stage of the program development life cycle.

.....

.....

.....

..... [2]

JUNE 2024 (22)

3 Identify **three** different tasks in the analysis stage of the program development life cycle.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

NOVEMBER 2024 (21)

4 Identify **three** stages of the program development life cycle from the following list of words.

analysis **decomposition** **design** **input**

pseudocode **testing** **variable**

1 .....

2 .....

3 .....

[3]

5 Describe **three** methods that are used to design and construct a solution to a problem.

Method 1 .....

.....

.....

.....

Method 2 .....

.....

.....

.....

Method 3 .....

.....

.....

.....

[6]

NOVEMBER 2024 (22)

5 Analysis is one stage in the program development life cycle.

(a) State **one** other stage in the program development life cycle.

..... [1]

(b) Describe the analysis stage of the program development life cycle.

.....

.....

.....

.....

.....

..... [3]

3 A program has been written to check the value of a measurement. The measurement must be a positive number and given to three decimal places, for example, `3.982`

(a) (i) State suitable examples of normal and erroneous test data that could be used to test this program. For each example give the reason for your choice of test data.

Normal test data example .....

Reason .....

.....

Erroneous test data example .....

Reason .....

.....

[4]

(ii) Explain why two pieces of boundary test data are required for this program.
Give an example of each piece of boundary test data.

.....

.....

.....

.....

..... [3]

(b) Explain why verification is needed and how verification could be performed by this program.

.....

.....

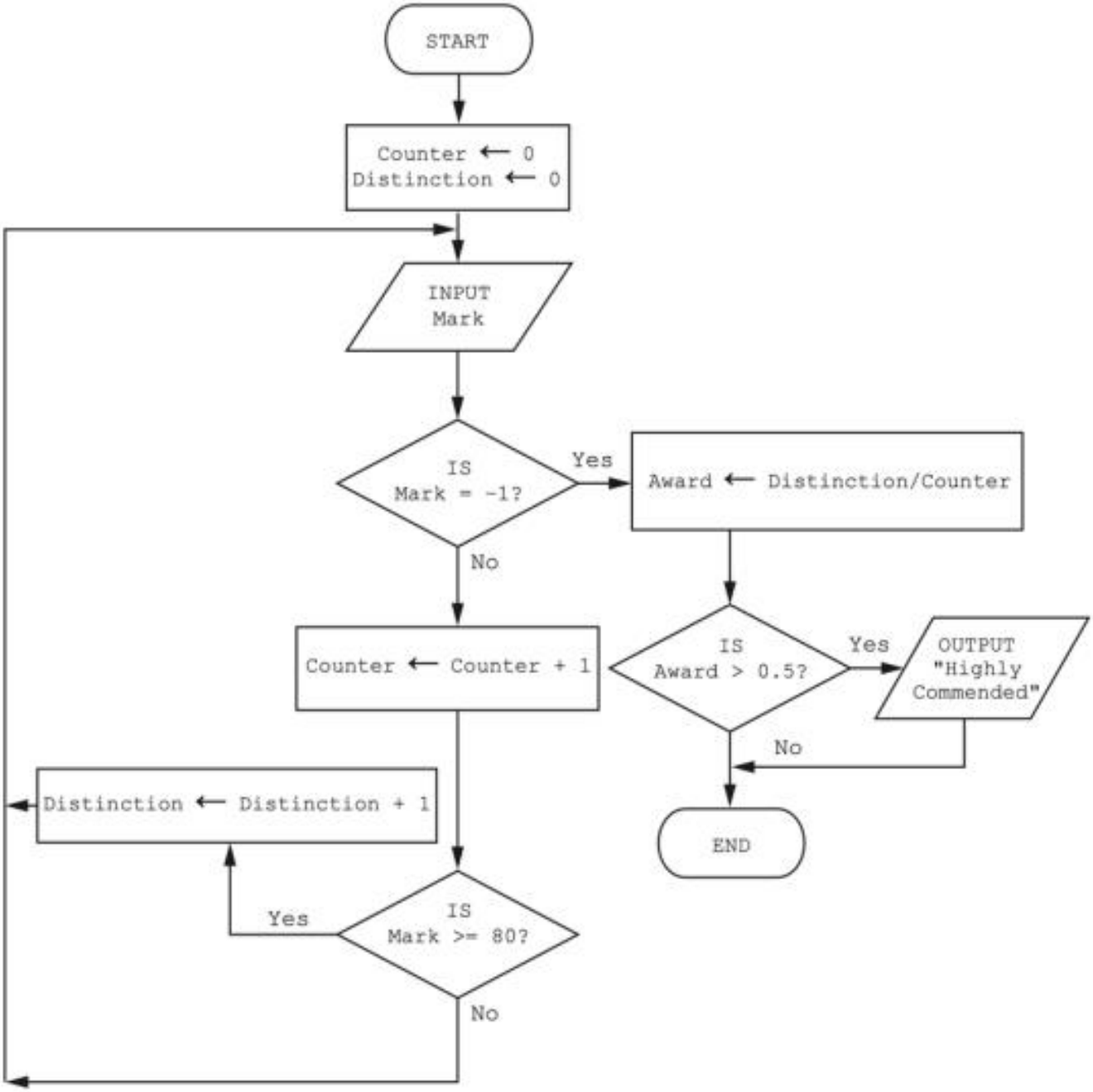
.....

.....

.....

..... [3]

4 The algorithm shown by this flowchart allows the input of examination marks for a class of students. A mark of –1 ends the process. If a mark is 80 or over then a distinction grade is awarded. The number of distinctions for the whole class is calculated. If this is over 50% of the class, the class is awarded a highly commended certificate.



Complete a trace table for the algorithm using this input data:
88, 74, 60, 90, 84, 87, 95, 72, 84, 66, –1

Counter	Distinction	Mark	Award	OUTPUT

[5]

NOVEMBER 2021 (2)

2 Tick (✓) one box in each row to identify if the statement is about validation, verification or neither.

Statement	Validation (✓)	Verification (✓)	Neither (✓)
a check where data is re-entered to make sure no errors have been introduced during data entry			
an automatic check to make sure the data entered has the correct number of characters			
a check to make sure the data entered is sensible			
a check to make sure the data entered is correct			

[3]

3 A program checks that the data entered is between 1 and 100 inclusive.

Identify **one** piece of normal, extreme and erroneous test data for this program, and give a reason for each.

Normal test data .....

Reason .....

.....

.....

Extreme test data .....

Reason .....

.....

.....

Erroneous test data .....

Reason .....

.....

.....

[6]

JUNE 2022 (1)

2 Tick (✓) **one** box in each row to identify the most appropriate data type for each description. Only **one** tick (✓) per column.

Description	Data type				
	Boolean	Char	Integer	Real	String
a single character from the keyboard					
multiple characters from the keyboard					
only one of two possible values					
only whole numbers					
any number					

[4]

3 Give **one** piece of normal test data and **one** piece of erroneous test data that could be used to validate the input of an email address.

State the reason for your choice in each case.

Normal test data .....

.....

Reason .....

.....

.....

.....

Erroneous test data .....

.....

Reason .....

.....

.....

.....

[4]

JUNE 2022 (3)

2 Tick (✓) **one** or more boxes in each row to match the type(s) of test data to each description.

Description	Types of test data			
	Boundary	Erroneous / Abnormal	Extreme	Normal
test data that is always on the limit of acceptability				
test data that is either on the limit of acceptability or test data that is just outside the limit of acceptability				
test data that will always be rejected				
test data that is within the limits of acceptability				

[4]

3 Describe what is meant by the terms variable and constant and give an example of each in your answer.

Variable .....

.....

.....

.....

Constant .....

.....

.....

.....

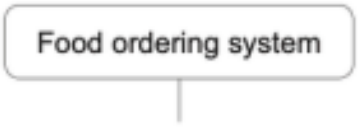
[4]

MARCH 2023

5 A food ordering system is an example of a computer system that is made up of sub-systems.

The food ordering system:
- allows the user to enter the details of the food they want to order and to pay for the order
- displays food available as pictures or as a list.

Complete the structure diagram for the given parts of the food ordering system.



[4]

NOVEMBER 2024 (23)

10 An algorithm has been written in pseudocode to check that a password meets a set of rules.

```
01 OUTPUT "Please enter password "
02 INPUT Password
03 Accept ← TRUE
04 IF LENGTH(Password) < 8 OR LENGTH(Password) > 20
05   THEN
06     Accept ← FALSE
07 ENDIF
08 IF LCASE(Password) = Password OR UCASE(Password) = Password
09   THEN
10     Accept ← FALSE
11 ENDIF
12 Index ← 1
13 Found ← FALSE
14 WHILE NOT Found AND Accept AND Index < LENGTH(Password)
15     IF SUBSTRING(Password, Index, 1) = '!'
16       THEN
17         Found ← TRUE
18     ENDIF
19     Index ← Index + 1
20 ENDWHILE
21 IF NOT Found
22   THEN
23     Accept ← FALSE
24 ENDIF
25 IF Accept
26   THEN
27     OUTPUT "Accepted"
28   ELSE
29     OUTPUT "Rejected"
30 ENDIF
```

(a) Complete the **three** trace tables using the data shown for each one.

Data: MYWORD

Password	Accept	Index	Found	OUTPUT

Data: M!word

Password	Accept	Index	Found	OUTPUT

Data: My!Hidden

Password	Accept	Index	Found	OUTPUT

[6]

(b) State the rules that the password must meet.

.....

.....

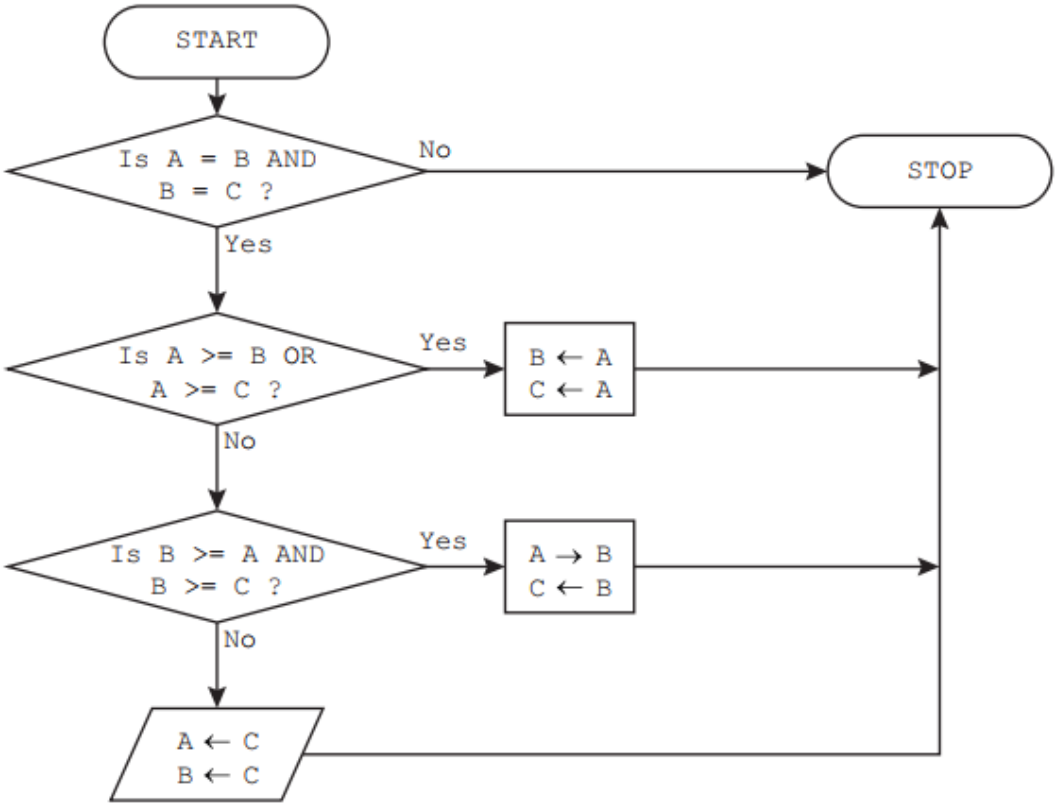
.....

.....

.....

..... [3]

9 A flowchart checks that values stored in three variables are identical.
If they are different, the highest value is stored in all three variables.



Identify **four** errors in the flowchart and suggest a correction for each error. You may label each error on the diagram with the corresponding error number.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

[4]

4 The pseudocode represents an algorithm.

The pre-defined function `DIV` gives the value of the result of integer division.
For example, `Y = 11 DIV 4` gives the value `Y = 2`

```
Count ← 0
INPUT Limit
FOR In ← 1 TO Limit
  Logic ← TRUE
  Test ← 2
  INPUT Number
  REPEAT
    IF Number / Test = Number DIV Test
      THEN
        Logic ← FALSE
      ELSE
        Test ← Test + 1
    ENDIF
  UNTIL NOT Logic OR Test >= Number DIV 2
  IF Logic
    THEN
      Store[Count] ← Number
      Count ← Count + 1
  ENDIF
NEXT In
FOR Out ← 0 TO Count - 1
  OUTPUT Store[Out]
NEXT Out
```

(a) Complete the trace table for the algorithm using this input data:
5, 9, 5, 8, 10, 7

In	Logic	Test	Number	Store [Count]	Count	Limit	Out	OUTPUT

[7]

(b) State the purpose of this algorithm.

.....

.....

.....

..... [2]

(c) This algorithm only works for numbers that are 3 or greater.

Describe how you could change this algorithm to make sure that only numbers that are 3 or greater are entered. Any pseudocode statements used in your answer must be fully described.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

JUNE 2023

3 **(a)** Describe what is meant by data validation.

.....

.....

.....

..... [2]

6 The energy efficiency of an electrical appliance is the percentage of useful energy out compared with the total energy in.

An algorithm has been written in pseudocode to calculate the energy efficiency of an appliance. Values for total energy in and useful energy out are input. The efficiency is calculated and output as a percentage.
The entry of the number –1 for either value stops the algorithm.

```
01 REPEAT
02     OUTPUT "Enter total energy in "
03     INPUT TotalEnergyIn
04     OUTPUT "Enter useful energy out "
05     OUTPUT UsefulEnergyOut
06     IF TotalEnergyIn <> -1 AND UsefulEnergy <> -1
07       THEN
08         Efficiency ← (UsefulEnergyOut / TotalEnergyIn) * 100
09         OUTPUT "Efficiency is ", Efficiency, "%"
10     ENDIF
11 UNTIL TotalEnergyIn <> -1 OR UsefulEnergyOut <> -1
```

(a) Identify the **three** errors in the pseudocode and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

[3]

(b) Write pseudocode to check for an efficiency of 92% or over for this appliance and to output "A-rated" if the efficiency is 92% or over.

.....

.....

.....

.....

..... [2]

4 A program needs to make sure the value input for a measurement meets the following rules:
- the value is a positive number
- a value is always input
- the value is less than 1000.

(a) Describe the validation checks that the programmer would need to use.

.....

.....

.....

.....

.....

..... [3]

(b) The program needs editing to include a double entry check for the value input.

(i) State why this check needs to be included.

.....

..... [1]

(ii) The input value needs to be stored in the variable `Measurement`
Write pseudocode to perform the double entry check until a successful input is made.

.....

.....

.....

.....

.....

.....

.....

..... [3]

5 (a) **Four** descriptions of validation checks are shown.

Draw **one** line to link each description to the most appropriate check.

Not all checks will be used.

Description	Check
to check that the data entered is an integer	check digit
to check that some data has been entered	format check
to check that the data entered has an appropriate number of characters	length check
to check that an identification number contains no errors	presence check
	type check

[4]

(b) Write an algorithm in pseudocode to make sure that an input for the variable `Length` is between 15 and 35 inclusive. The code must iterate until a valid input has been made and the code must include appropriate messages.

.....

.....

.....

.....

.....

.....

.....

..... [3]

7 An algorithm has been written in pseudocode to calculate a check digit for a four-digit number. The algorithm then outputs the five-digit number including the check digit.
The algorithm stops when –1 is input as the fourth digit.

```
01 Flag ← FALSE
02 REPEAT
03     Total ← 0
04     FOR Counter ← 1 TO 4
05       OUTPUT "Enter a digit ", Counter
06       INPUT Number[Counter]
07       Total ← Total + Number * Counter
08       IF Number[Counter] = 0
09         THEN
10           Flag ← TRUE
11       ENDIF
12     NEXT Counter
13     IF NOT Flag
14       THEN
15         Number[5] ← MOD(Total, 10)
16         FOR Counter ← 0 TO 5
17             OUTPUT Number[Counter]
18         NEXT
19     ENDIF
20 UNTIL Flag
```

(a) Give the line number(s) for the statements showing:

Totalling .....

Count-controlled loop .....

Post-condition loop .....

[3]

(b) Identify the **three** errors in the pseudocode and suggest a correction for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

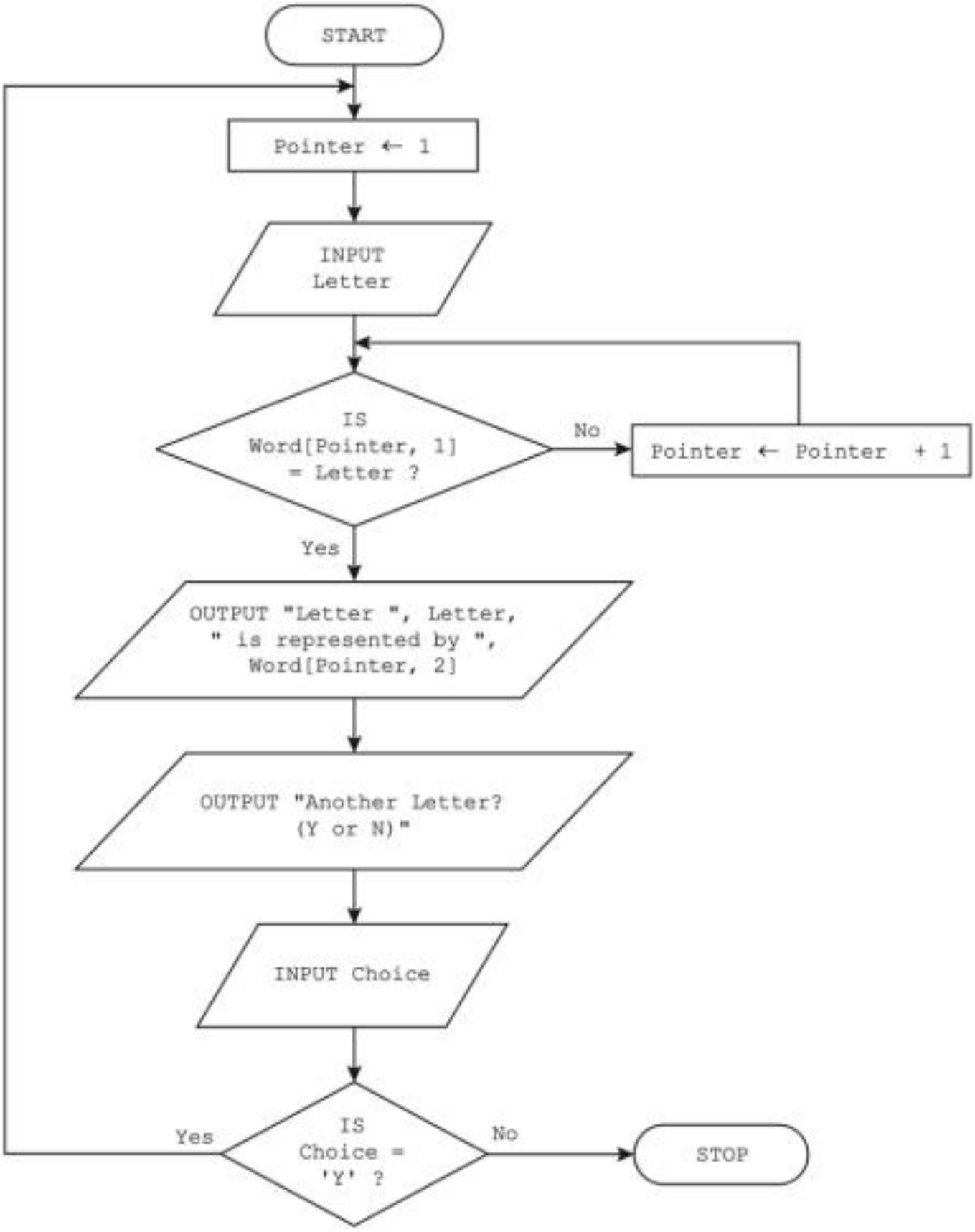
Error 3 .....

Correction .....

.....

[3]

7 The flowchart represents an algorithm.



The table represents the two-dimensional (2D) array `Word[]` which stores the first half of the phonetic alphabet used for radio transmission. For example, `Word[10,1]` is 'J'.

Index	1	2
1	A	Alpha
2	B	Bravo
3	C	Charlie
4	D	Delta
5	E	Echo
6	F	Foxtrot
7	G	Golf
8	H	Hotel
9	I	India
10	J	Juliet
11	K	Kilo
12	L	Lima
13	M	Mike

(a) Complete the trace table for the algorithm by using the input data: F, Y, D, N

Pointer	Letter	Choice	OUTPUT

[4]

(b) Identify the type of algorithm used.

.....

..... [1]

(c) Describe **one** problem that could occur with this algorithm if an invalid character was input.

.....

.....

.....

..... [2]

SPECIMEN 2023 – 2A

8 A programmer has written an algorithm to check that prices are less than $10.00

These values are used as test data:

10.00 9.99 ten

State why each value was chosen as test data.

10.00 .....

.....

9.99 .....

.....

ten .....

.....

[3]

8 A programmer is designing a program to check the length of a password and to check if the password input is the same as the stored password.

The program requirements are:

- input the password, `Password`
- check if there are at least 8 characters in the password
- check that the password is **not** the same as the stored password `OldPass`
- output 'accepted' if both tests are completed successfully
- otherwise, output 'rejected'.

Use the variable names given.

(a) Complete the flowchart for the program.

[6]

5 An algorithm has been written in pseudocode to calculate the profit when an item is sold. Values for cost price and selling price are input, the profit is calculated (selling price – cost price) and output. The input of zero for either value stops the algorithm.

```
01 REPEAT
02      OUTPUT "Enter cost price "
03      INPUT Cost
04      OUTPUT "Enter selling price "
05      OUTPUT Sell
06      IF Cost <> 0 OR Sell <> 0
07        THEN
08          Profit ← Sell - Cost
09          OUTPUT "Profit is ", Profit
10      NEXT
11 UNTIL Cost = 0 OR Sell = 0
```

(a) Identify the line numbers of **three** errors in the pseudocode and suggest corrections.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

[3]

(b) This algorithm needs to be improved with a range check.

(i) Write pseudocode to reject the input of values less than zero for variables `Cost` and `Sell`

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(ii) Describe **two** other validation checks that should be included in this algorithm.

Check 1 .....

.....

.....

.....

Check 2 .....

.....

.....

.....

[4]

3 A linear search and a bubble sort are standard methods of solution.
Fifty numbers are already stored in the array `Values[1:50]`

(a) Write an algorithm in pseudocode to input a number, `MyNumber`, and use a linear search to test if that number is stored in the array. If the number is found in the array, the position in the array is output. If the number is **not** found in the array, "Not found" is output.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) Write an algorithm in pseudocode to sort the array `Values[1:50]` into ascending order using a bubble sort.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

4 Identify **three** data types used in programming.

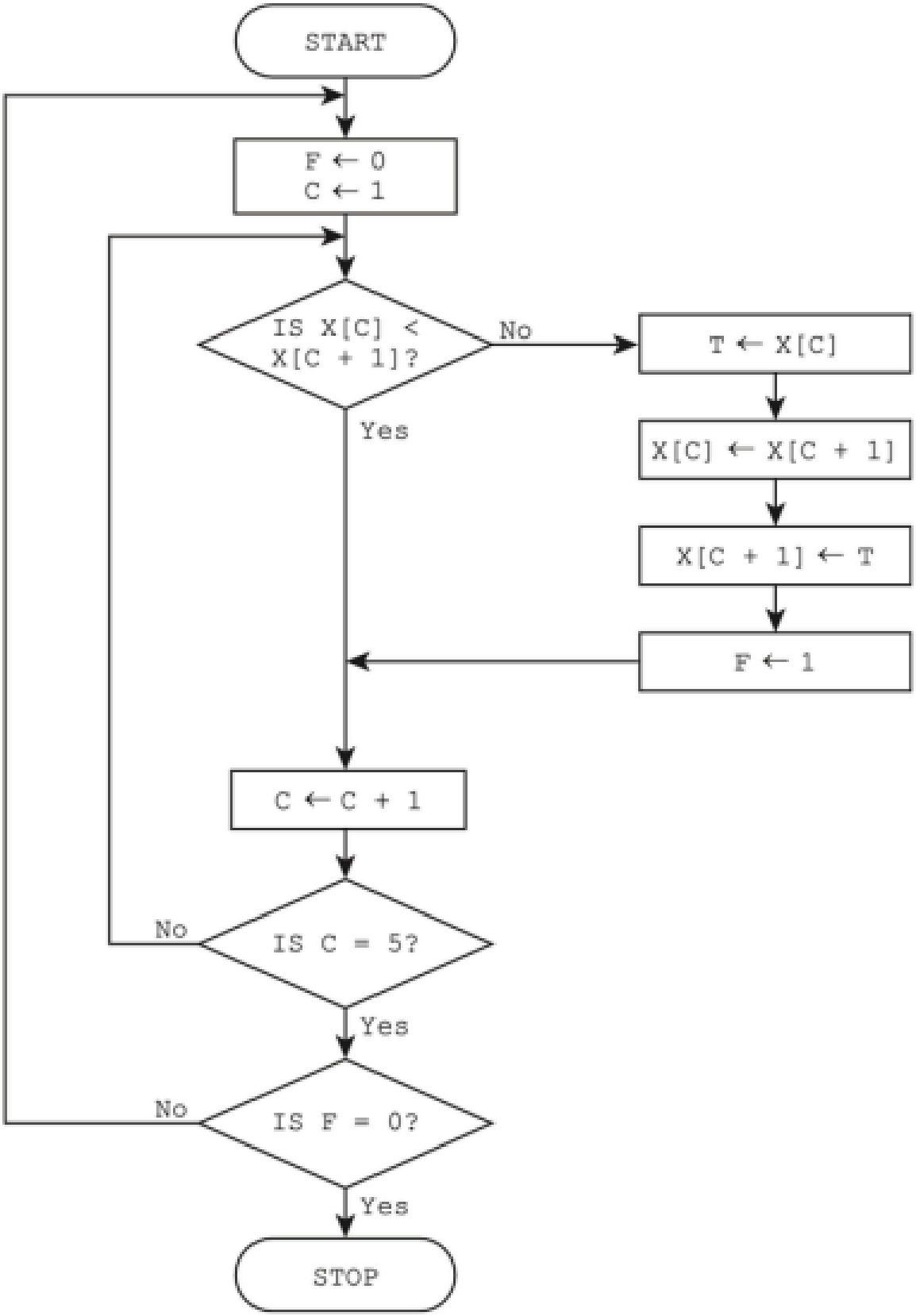
1 .....

2 .....

3 .....

[3]

9 This flowchart represents an algorithm.



(a) The array `X[1:5]` used in the flowchart contains this data:

X[1]	X[2]	X[3]	X[4]	X[5]
10	1	5	7	11

Complete the trace table by using the data given in the array.

F	C	X[1]	X[2]	X[3]	X[4]	X[5]	T
		10	1	5	7	11	

[5]

(b) Describe what the algorithm represented by the flowchart is doing.

.....

.....

.....

..... [2]

JUNE 2023 (3)

4 **(a)** Explain why verification checks are used when data is input.

.....

.....

.....

..... [2]

(b) Give **two** types of verification check and state how each one can be used.

Verification check 1 .....

Use .....

.....

Verification check 2 .....

Use .....

.....

[4]

NOVEMBER 2023 (2)

2 A type of validation check is a length check. Another type of validation check is used to make sure that any date entered is in the dd/mm/yyyy style:
dd means day, mm means month and yyyy means year.

(a) State the type of validation check used.

..... [1]

(c) Describe how a length check could be used with the date entered.

.....

.....

.....

..... [2]

6 An algorithm has been written in pseudocode.

```
01 DECLARE A[1:10] : STRING
02 DECLARE T : STRING
03 DECLARE C, L : INTEGER
04 L ← 10
05 FOR C ← 1 TO L
06     OUTPUT "Please enter name "
07     INPUT A[C]
08 NEXT C
09 FOR C ← 1 TO L
10     FOR L ← 1 TO 9
11         IF A[L] > A[L + 1]
12           THEN
13             T ← A[L]
14             A[L] ← A[L + 1]
15             A[L + 1] ← T
16         ENDIF
17     NEXT L
18 NEXT C
19 FOR C ← 1 TO L
20     OUTPUT "Name ", C, " is ", A[C]
21 NEXT C
```

(a) State the purpose of this pseudocode algorithm.

.....

..... [1]

(b) State **four** processes in this algorithm.

1 .....

.....

2 .....

.....

3 .....

.....

4 .....

.....

[4]

(c) Meaningful identifiers have **not** been used in this algorithm.
Suggest suitable meaningful identifiers for:

The array:

A .....

The variables:

T .....

C .....

L .....

[3]

(d) State **two** other ways the algorithm can be made easier to understand and maintain.

1 .....

.....

2 .....

.....

[2]

8 A programmer is designing an algorithm to calculate the cost of a length of rope.
The program requirements are:

- input two values: the length of rope in metres `Length` and the cost of one metre `Cost`
- perform a validation check on the length to ensure that the value is between `0.5` and `6.0` inclusive
- calculate the price `Price`
- output the price rounded to two decimal places.

Use the variable names given.

(a) State the name of the validation check.

..... [1]

(b) Complete the flowchart for this algorithm.

[6]

(c) Give **two** different sets of test data for this algorithm and state the purpose of each set.

Set 1 .....

Purpose .....

.....

.....

Set 2 .....

Purpose .....

.....

.....

[4]

(d) Complete the headings for the trace table to show a dry-run for this algorithm.
You do **not** need to trace the algorithm.

.....	.....	.....	.....

[3]

(e) Describe an improvement that should be made to the requirements for this algorithm.

.....

.....

.....

..... [2]

4 This pseudocode algorithm is intended to allow data for up to 50 people to be entered and stored in a two-dimensional (2D) array. The data is their last name, first name and the city in which they live.

```
01 DECLARE People : ARRAY[1:50, 1:3] OF REAL
02 DECLARE Count : INTEGER
03 DECLARE Response : CHAR
04 DECLARE Continue : BOOLEAN
05 FOR I ← 1 TO 50
06     FOR J ← 1 TO 3
07         People[I, J] ← ""
08     NEXT J
09 NEXT I
10 Count ← 100
11 Continue ← TRUE
12 CASE OF
13     OUTPUT "Enter the last name"
14     INPUT People[Count, 1]
15     OUTPUT "Enter the first name"
16     INPUT People[Count, 2]
17     OUTPUT "Enter the city"
18     INPUT People[Count, 3]
19     OUTPUT "Do you want to enter another name (Y or N)?"
20     INPUT Response
21     IF Response = 'N'
22       THEN
23         Continue ← FALSE
24       ELSE
25         Count ← Count + 1
26     ENDIF
27 UNTIL NOT Count
```

(a) Identify the line numbers of the **four** errors in the pseudocode and suggest corrections.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

Error 4 line number .....

Correction .....

.....

[4]

(b) Write the pseudocode that you could add to the end of this algorithm to output the contents of the array. Make sure that the output ends when the data in the array ends.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Explain how you could alter the original corrected algorithm to make sure that the number of elements being added to the array does not exceed the maximum size of the array (50 elements).

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) An algorithm needs to total 50 numbers between 1 and 100 inclusive.

Draw a flowchart that:

- uses a count-controlled loop from 1 to 50
- uses an appropriate prompt to ask for a number between 1 and 100
- totals the numbers as they are entered
- outputs the total after the loop has completed with an appropriate message.

[6]

4 This pseudocode algorithm is intended to sort a pre-populated one-dimensional (1D) array named `ItemList` into alphabetical order using a bubble sort.

```
01 DECLARE ItemList : ARRAY[1:100] OF STRING
02 DECLARE Counter : STRING
03 DECLARE Limit : INTEGER
04 DECLARE Pass : INTEGER
05 DECLARE Swapped : BOOLEAN
06 DECLARE Temp : STRING
07 Limit ← 100
08 Pass ← 1
09 Temp ← TRUE
10 WHILE Swapped = TRUE OR Pass <= Limit - 1 DO
11     Swapped ← FALSE
12     FOR Counter ← 1 TO Limit - Pass
13         IF ItemList[Counter] > ItemList[Counter + 1]
14           THEN
15             Temp ← ItemList[Counter]
16             ItemList[Counter] ← ItemList[Counter + 1]
17             ItemList[Counter] ← Temp
18             Swapped ← TRUE
19         ENDCASE
20         Pass ← Pass + 1
21     NEXT Counter
22 ENDWHILE
```

(a) Identify the line numbers of **five** errors in the pseudocode and 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 .....

.....

Error 5 line number .....

Correction .....

.....

[5]

(b) A bubble sort algorithm can be written to include features that make it more efficient.

Explain why the **corrected** bubble sort algorithm is efficient.

.....

.....

.....

.....

.....

..... [3]

4 **(a)** State **one** reason for using a verification check when inputting data.

.....

..... [1]

(b) Write the pseudocode for an algorithm to perform a double entry verification check, using the following criteria:

- Input 10 numbers and store them in the array `Numbers[]`
- Input the same 10 numbers again; as each number is input, compare it with its corresponding number in the array `Numbers[]`
  - if the numbers match, move on to the next number
  - if the numbers do not match, display both numbers and ask for the number to be re-entered
  - store the re-entered number in the corresponding array location and move on to the next number.
- When all the numbers have been checked, display a message to say the check has been completed.

You do **not** need to declare any arrays or variables for this algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

5 This pseudocode algorithm should allow 500 names of people and their heights to be entered and stored in two one-dimensional (1D) arrays. The name of a person and the height of that person are in the same index in both the `Names[]` and the `Heights[]` arrays.

The algorithm performs as follows:

- at the start, the following data is entered:
  - name of person
  - height of person in cm
- the data is stored in the relevant arrays
- the heights of the people are totalled
- the shortest person is found
- at the end, the following is output:
  - name of shortest person
  - height of shortest person
  - average height of all 500 people.

```
01  DECLARE Names : ARRAY[1:500] OF REAL
02  DECLARE Heights : ARRAY[1:500] OF REAL
03  DECLARE Shortest : REAL
04  DECLARE Total : REAL
05  DECLARE Counter : INTEGER
06  DECLARE Index : INTEGER
07  Shortest ← 500
08  Index ← 0
09  Total ← 100
10  FOR Counter ← 1 TO 500
11      INPUT Names[Index]
12      INPUT Heights[Counter]
13      Total ← Counter + Heights[Counter]
14      IF Heights[Counter] < Shortest
15        THEN
16          Shortest ← Heights[Counter]
17          Index ← Counter
18      ENDIF
19  NEXT Counter
20  OUTPUT "The shortest height is ", Heights
21  OUTPUT "The shortest person is ", Names[Index]
22  OUTPUT "The average height is ", Total / 500
```

(a) Identify the line numbers of **five** errors in the pseudocode and 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 .....

.....

Error 5 line number .....

Correction .....

.....

[5]

(b) Write the pseudocode statement that will output the average height rounded to **one** decimal place.

.....

..... [2]

(c) Explain how you could change the corrected algorithm so that it will also:

- find the height of the tallest person.
- output the height of the tallest person at the end of the algorithm.

Any program statements used must be fully explained.

.....

.....

.....

.....

.....

.....

.....

..... [4]