Past Papers May/June 2015 to 2018, Oct/Nov 2015 till 2024:
Section P2: Pseudocodes, Flowcharts & Programming concepts

(2210/21/M/J/15)

Q1/ 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.

1...

....

2...

....

3..

....

4...

....[4]

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

....

....

....[2]

3. Identify three different loop structures that you can use when writing pseudocode.

1..

2..

3.. [3]

4/- Five data types and five data samples are shown below.
Draw a line to link each data type to the correct data sample.

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

[4]

(2210/21/M/J/15)
5/- Read this section of program code that should input 30 positive numbers and then output the largest number input.

```
Large = 9999
Counter = 0
WHILE Counter > 30
  DO
     INPUT Num
     IF Num < Large THEN Large = Num
     Counter = Counter - 1
ENDWHILE
PRINT Large
```

There are four errors in this code.
Locate these errors and suggest a corrected piece of code for each error.

1...

2...

3...

4...[4]

6/- Four programming concepts and four examples of programming code are shown below.
Draw a line to link each programming concept to the correct example of programming code.

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

[4]

(2210/21/M/J/16)

7/- Read this section of program code that inputs 10 positive numbers and then outputs the smallest number input.

```
1     Small = 1000
2     Counter = 0
3     REPEAT
4     INPUT Num
5     IF Num < Small THEN Small = Num
6     Counter = Counter + 1
7     UNTIL Counter = 10
8     PRINT Small
```

(i) Identify three changes you would need to make to find the largest number input instead of the smallest number.

1 ....

....

2 ....

....

3 ....

....[3]

(ii) Rewrite the program code with your changes.

....

....

....

....

....

....

....

....[3]

8/- A program will be written to store information about members of a swimming club.
The following membership details will be recorded:

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

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

Membership details	Data type
Name	
Gender	
Status	
Fee	
Team member	

[5]

(ii) The swimming club has 50 members.

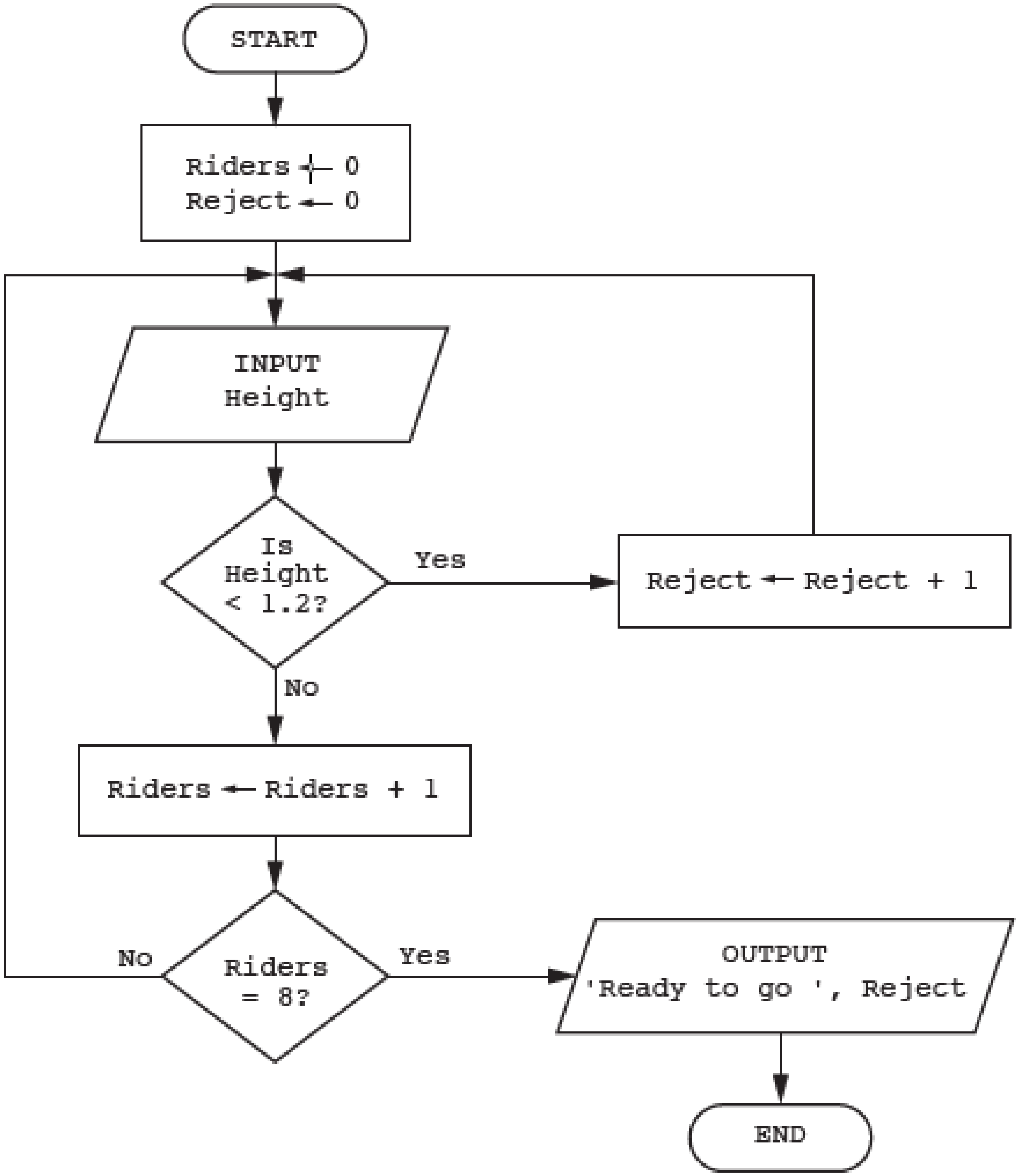
State the data structure that would be most suitable to use and give a reason for your choice.

Data structure ....

Reason ....

....[2]

9 The flowchart below inputs the height of children who want to ride on a rollercoaster. Children under 1.2 metres are rejected. The ride starts when eight children have been accepted.



Complete the trace table for the input data:

1.4, 1.3, 1.1, 1.3, 1.0, 1.5, 1.2, 1.3, 1.4, 1.3, 0.9, 1.5, 1.6, 1.0

Riders	Reject	Height	OUTPUT

[4]

10 `REPEAT ... UNTIL` is one type of loop structure.

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

Loop structure 1 ....

Description ....

....

Loop structure 2 ....

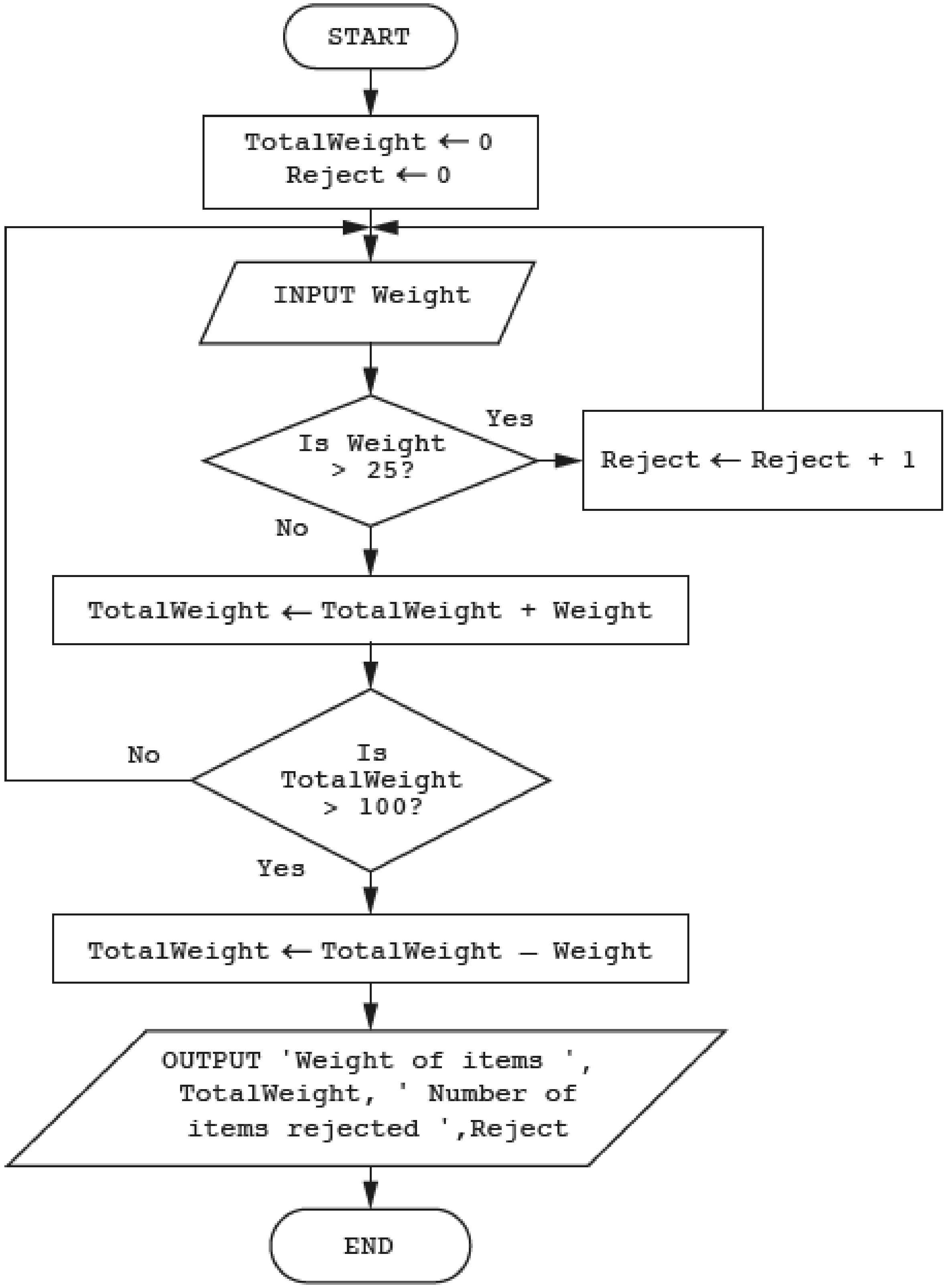
Description ....

....[4]

2210/22/M/J/17

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

11b This flowchart inputs the weight of items in kilograms to be loaded on a trailer.
Any item over 25 kilograms is rejected.
The trailer can take up to 100 kilograms.


START
TotalWeight ← 0
Reject ← 0
INPUT Weight
Is Weight > 25?
Yes
Reject ← Reject + 1
No
TotalWeight ← TotalWeight + Weight
No
Is TotalWeight > 100?
Yes
TotalWeight ← TotalWeight – Weight
OUTPUT 'Weight of items ', TotalWeight, ' Number of items rejected ',Reject
END

Complete the trace table for the input data:
13, 17, 26, 25, 5, 10, 15, 35, 20, 15

Welght	Reject	TotalWelght	OUTPUT

[5]

12 An algorithm has been written in pseudocode 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 pseudocode and suggest a correction.

Error ....

Correction ....

....[2]

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

....

....

....

....

....

....

....[3]

2210/22/M/J/18
13 (a) Draw a flowchart for an algorithm to input numbers. Reject any numbers that are negative and count how many numbers are positive. When the number zero is input, the process ends and the count of positive numbers is output.

[6]

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

....

....

....

....[2]

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

```
INPUT Number1, Number2, Sign

WHILE Number1 <> 0
     IF Sign = '+' THEN Answer Number1 + Number2 ENDIF
     IF Sign = '-' THEN Answer Number1 - Number2 ENDIF
     IF Sign = '*' THEN Answer Number1 * Number2 ENDIF
     IF Sign = '/' THEN Answer Number1 / Number2 ENDIF
     IF Sign <>'/' AND Sign <>'*' AND Sign <>'-' AND Sign <>'+'
          THEN Answer = 0
     ENDIF
     IF Answer <> 0
          THEN OUTPUT Answer
     ENDIF
     INPUT Number1, Number2, Sign
ENDWHILE
```

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

Number 1	Number2	Sign	Answer	OUTPUT

[3]

(b) Show how you could improve the algorithm written in pseudocode by writing an alternative type of conditional statement in pseudocode.

....

....

....

....

....

....

....

....

....

....[3]

15 A programmer has written a routine to store the name, email address and password of a contributor
to a website's discussion group.
(a) The programmer has chosen to verify the name, email address and password.
Explain why verification was chosen and describe how the programmer would verify this data.

....

....

....

....

....

....

....[4]

(b) The programmer has also decided to validate the email address and the password.
Describe validation checks that could be used.

Email address ....

....

....

Password ....

....

....[2]

16 A program checks that the weight of a basket of fruit is over 1.00 kilograms and under 1.10 kilograms. Weights are recorded to an accuracy of two decimal places and any weight not in this form has already been rejected.
Give **three** weights as **test data** and for each weight state a reason for choosing it.
All your reasons must be different.

Weight 1:
....
Reason
....

....
.
Weight 2:
....
Reason
:...

....

Weight 3:
....
Reason
:...

.... [3]

2210/22/M/J/19

17 (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`.

....

....

....

....

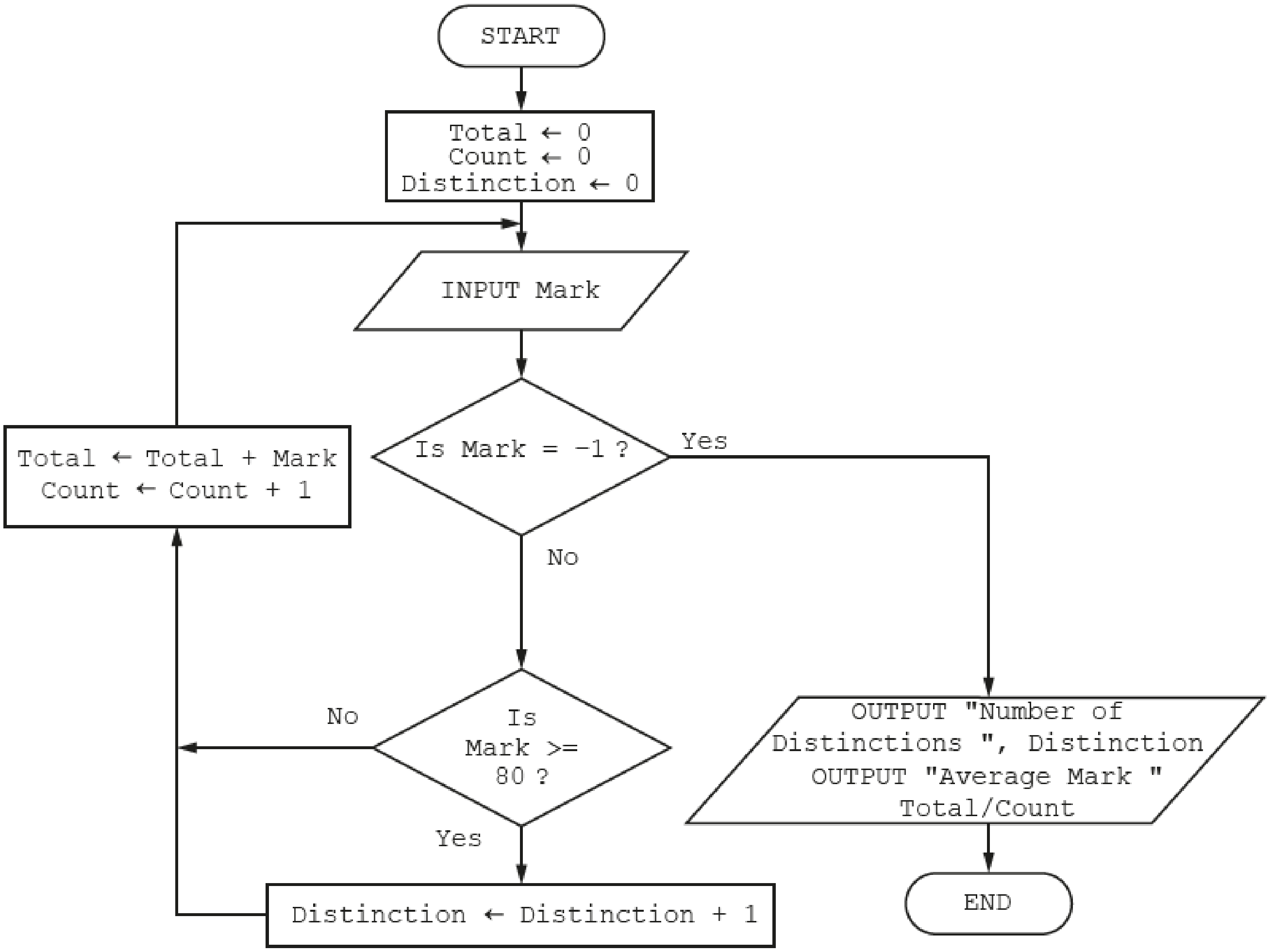
....

....

....

.... [4]

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



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

Total	Count	Distinction	Mark	OUTPUT

[4]

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

Statement	Selection	Repetition

`FOR A = 1 TO 100`
`B = B + 1`
`NEXT A`		

`CASE A OF`
`100: B = A`
`200: C = A`
`ENDCASE`		

`IF A > 100`
`THEN`
`B = A`
`ENDIF`		

`REPEAT`
`A = B * 10`
`UNTIL A > 100`		

[4]

(b) Explain what is meant by **validation** and **verification**.
Give an example for each one.

Validation

....

....

....

....

Example

....

....

....

Verification

....

....

....

....

Example

....

....

....[6]

20 (a)

Four pseudocode descriptions and five pseudocode statements are shown.

(a) Draw a line to link each pseudocode description to the most appropriate pseudocode statement.

Some pseudocode statements will **not** be used.

Pseudocode description

- a loop that will always iterate at least once
- a conditional statement to deal with many possible outcomes
- a loop that will always iterate a set number of times
- a conditional statement with different outcomes for true and false

Pseudocode statement

- FOR…TO…NEXT
- IF…THEN…ELSE…ENDIF
- WHILE…DO…ENDWHILE
- CASE…OF…OTHERWISE…ENDCASE
- REPEAT…UNTIL

[4]

(b) Using a single loop, write an algorithm in pseudocode to output 50 names that have been stored in the array, Name[ ]

....

....

....

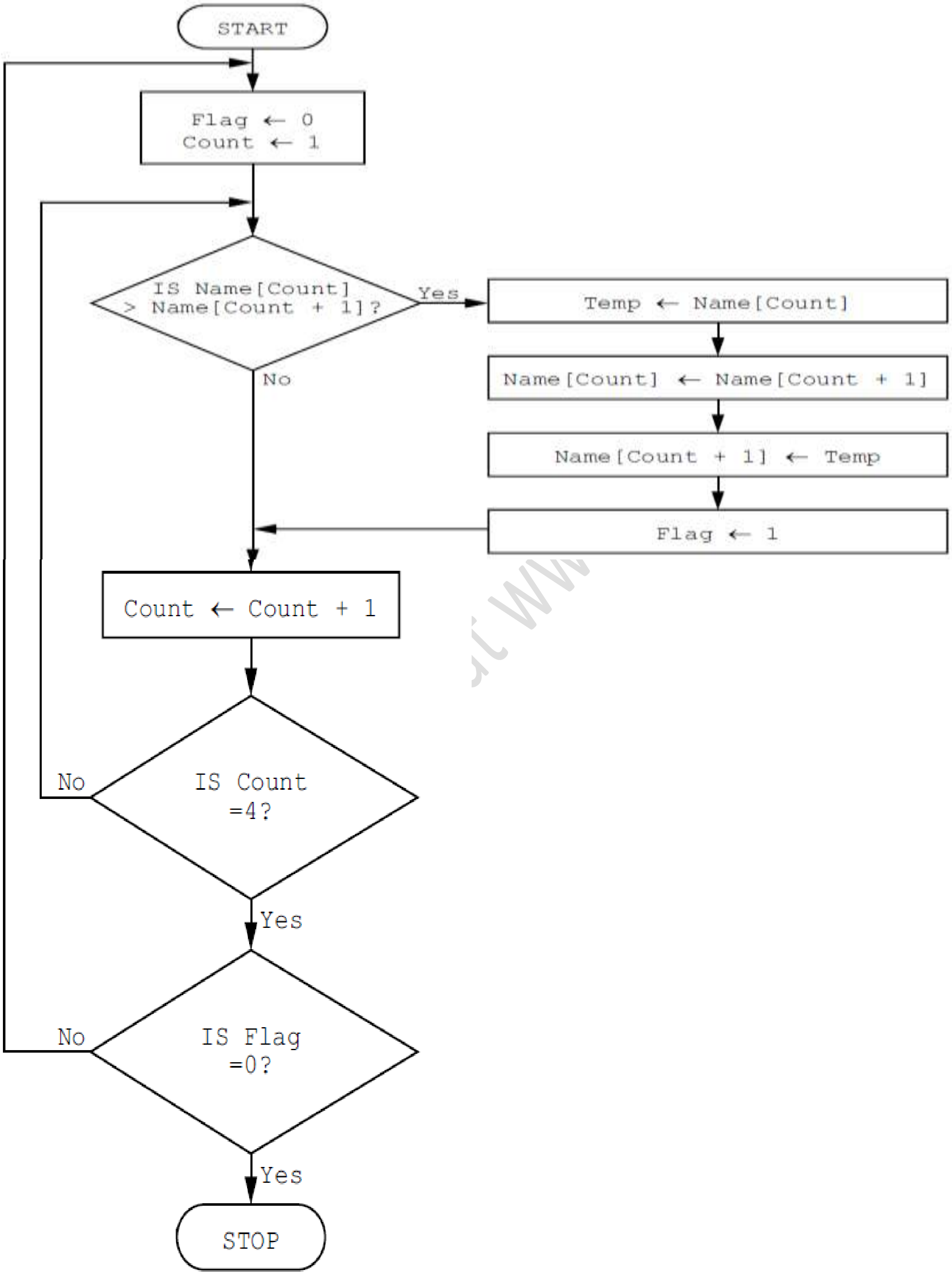
....

....

....[3]

2210/02/SP/23

21 (a



21 (a) The array `Name[1:4]` used in the flowchart contains the following data:

Name[1]	Name[2]	Name[3]	Name[4]
Jamal	Amir	Eve	Tara

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

Flag	Count	Name[1]	Name[2]	Name[3]	Name[4]	Temp
		Jamal	Amir	Eve	Tara	

[5]

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

....

....

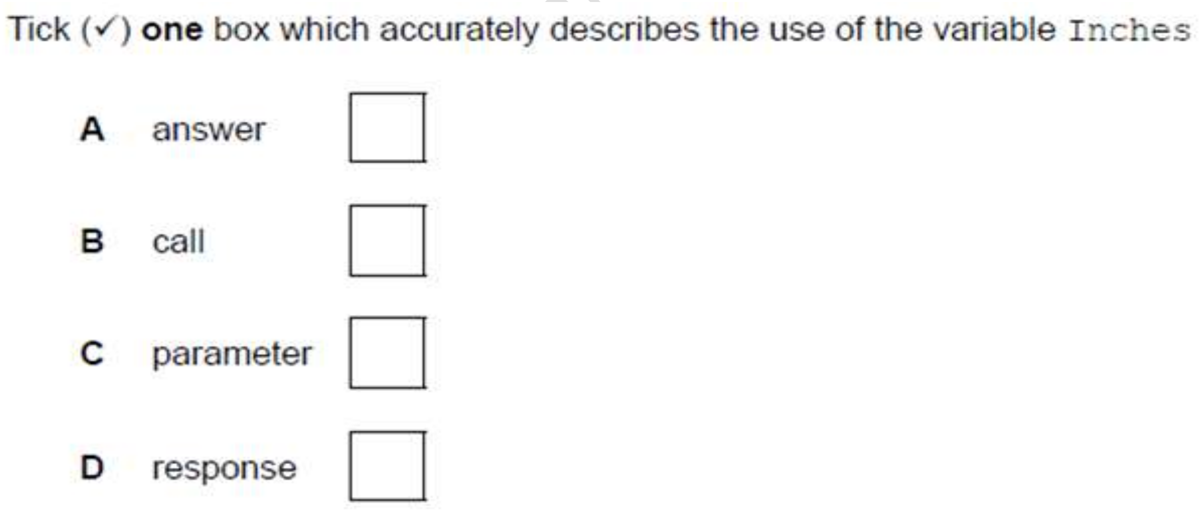
....

.... [2]

2210/02/SP/23

\22 A function is declared using pseudocode.

```
FUNCTION ConvertToCm(Inches: REAL) RETURNS REAL
         RETURN Inches * 2.4
ENDFUNCTION
```



Tick (✓) **one** box which accurately describes the use of the variable `Inches`

A answer □

B call □

C parameter □

D response □

(MARKING KEY)

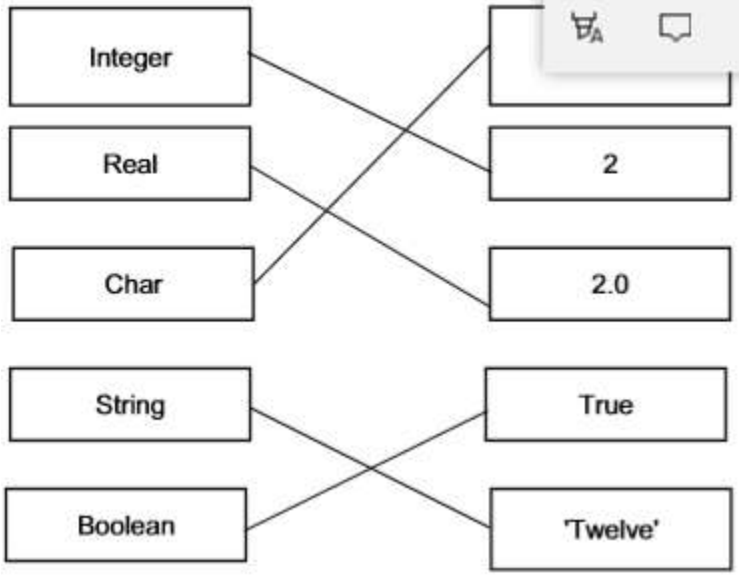
1/- mark for each error identified + suggested correction

- **Line 1** or Small = 0: this should read **Small = 999**
- **line 5** or IF…: this should read **IF Num < Small THEN Small = Num**
- **line 8** or UNTIL: this should read **UNTIL Counter = 10** or **UNTIL Counter > = 10** or **UNTIL Counter > 9**
- **line 7** or PRINT…: **PRINT** Small should come after the end of the repeat loop or line 8 or UNTIL: this should come before line 7 [4]

2/- Any two points from – a variable is used to store data that can change during the running of a program – a constant is used to store data that will not be changed during the running of a program [2] 3/-

- FOR (… TO … NEXT)
- REPEAT (… UNTIL)
- WHILE (… DO … ENDWHILE) [3]

4/- 1 mark for each correct link, up to maximum of 4 marks



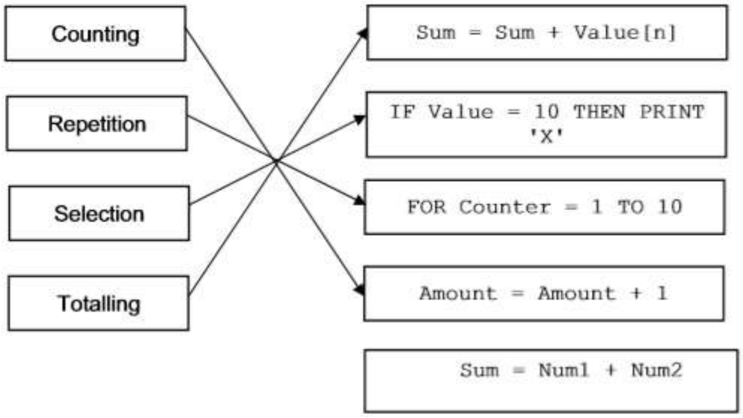
[4]

(2210/21/M/J/15)

5/- mark for each error identified + suggested correction.

- Line 1 or Large =9999: this should read Large = 0
- Line 3 or WHILE: this should read WHILE Counter < 30
- Line 6 or IF: this should read IF Num > Large THEN Large = Num
- Line 7 or Counter =…: this should read Counter = Counter + 1 [4]

6/- 1 mark for each correct line, two lines from one box not allowed



[4]

(2210/MJ/21/16)

7. (i) 1 mark for each change

- Change variable name in every instance as needs to be meaningful e.g. Large.
- Set this variable to a low value
- line 5: change comparison from < to > [3]

(ii) 3 marks maximum, 1 mark for each change correctly included.

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

[3]

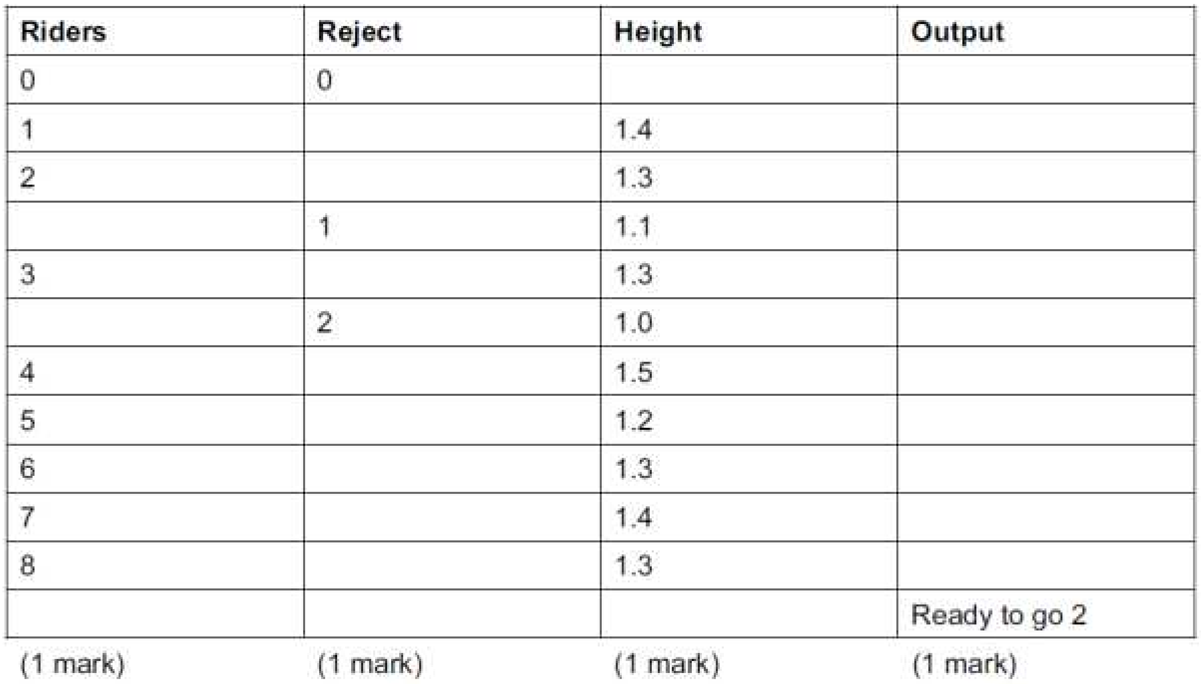
8 (i)

Name type – string
Gender type – char/string
Status type – char/string
Fee type – real
Team member type – Boolean [5]

8(ii) Data Structure – several Arrays ……

………….Reason – to simplify programming/ make programs shorter/index can be used to identify the same member across the arrays etc.

9



Riders	Reject	Height	Output
0	0		
1		1.4	
2		1.3	
	1	1.1	
3		1.3	
	2	1.0	
4		1.5	
5		1.2	
6		1.3	
7		1.4	
8		1.3	
			Ready to go 2
(1 mark)	(1 mark)	(1 mark)	(1 mark)

[4]

10– FOR (… TO … NEXT)…

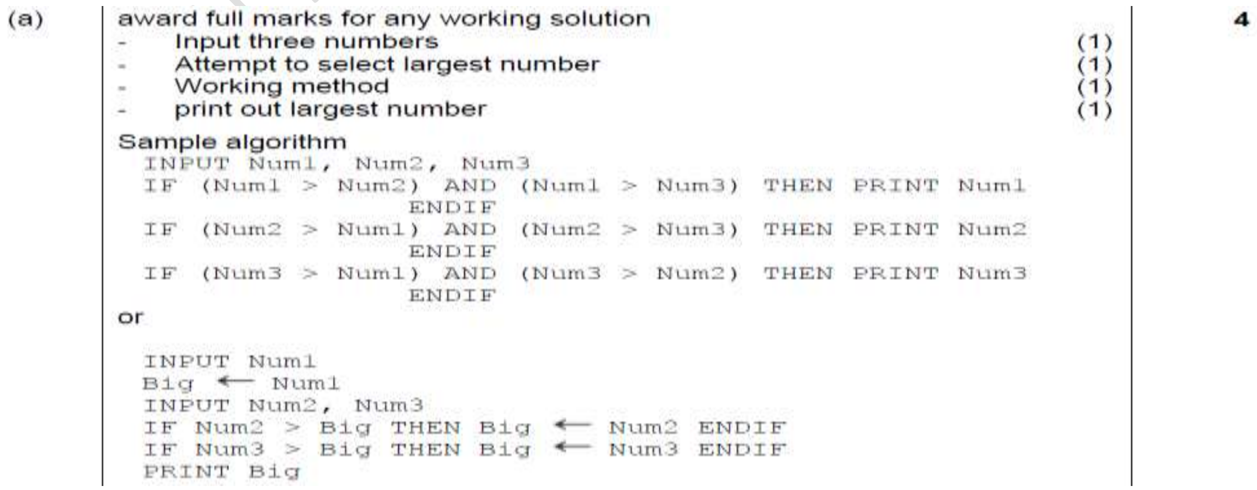
– … a set number of iterations

– WHILE (… DO … ENDWHILE) …

– … used where the loop may never be executed/whilst a specified condition exists

[4]

11



(a) award full marks for any working solution **4**

- Input three numbers (1)
- Attempt to select largest number (1)
- Working method (1)
- print out largest number (1)

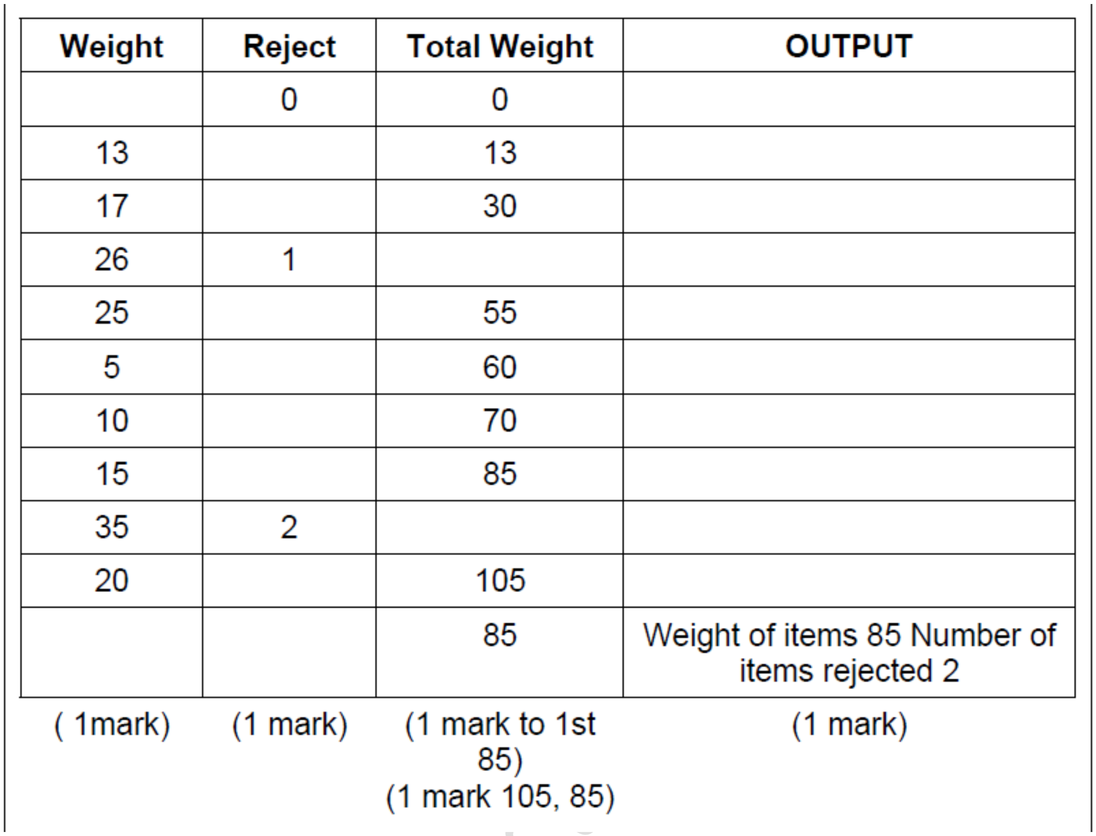
Sample algorithm

```
 INPUT Num1, Num2, Num3
 IF (Num1 > Num2) AND (Num1 > Num3) THEN PRINT Num1
          ENDIF
 IF (Num2 > Num1) AND (Num2 > Num3) THEN PRINT Num2
          ENDIF
 IF (Num3 > Num1) AND (Num3 > Num2) THEN PRINT Num3
          ENDIF
```

or

```
 INPUT Num1
 Big ← Num1
 INPUT Num2, Num3
 IF Num2 > Big THEN Big ← Num2 ENDIF
 IF Num3 > Big THEN Big ← Num3 ENDIF
 PRINT Big
```

11 (b)



Weight	Reject	Total Weight	OUTPUT
	0	0	
13		13	
17		30	
26	1		
25		55	
5		60	
10		70	
15		85	
35	2		
20		105	
		85	Weight of items 85 Number of items rejected 2
(1mark)	(1 mark)	(1 mark to 1st 85) (1 mark 105, 85)	(1 mark)

5

12(a)

Error - `Count 0`

Correction - `Count 1`

or

Error - `UNTIL Count > 100`

Correction - `UNTIL Count >= 100` or `UNTIL Count = 100`

or

`UNTIL Count > 99`

2

12(b)

– □ use of `FOR` with correct start and end values

- □ use of `NEXT`

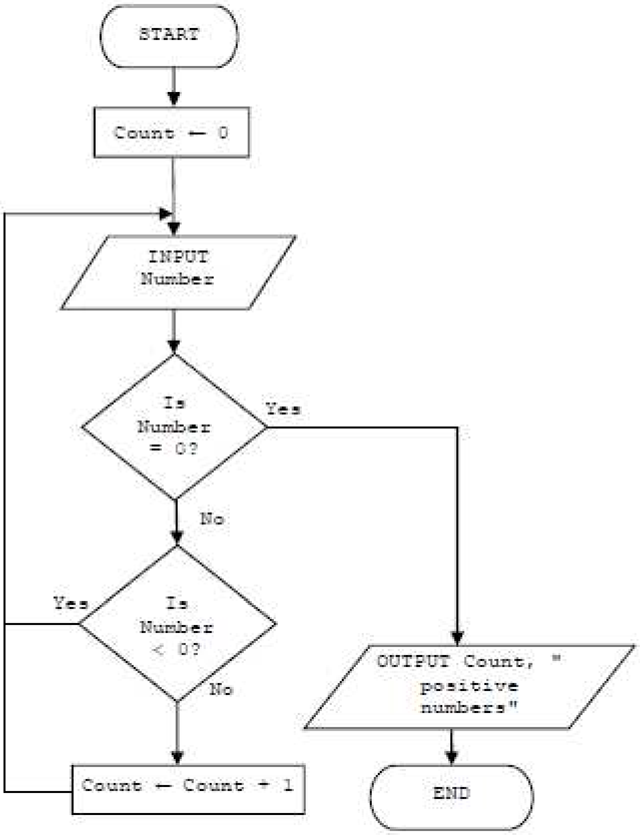
- □ removal of increment for `Count`

Sample algorithm

```
Sum 0
FOR Count 1 TO 100
INPUT Number
Sum Sum + Number
NEXT // NEXT Count
PRINT Sum
```

3

13 (a)

(a)	One mark per correct pair of actions, process, Input/Output, Tests (apart from START and END) max 3 One mark complete Flowlines, one mark working flowlines, one mark correct use flowchart symbols  START Count ← 0 INPUT Number Is Number = 0? Yes No Is Number < 0? Yes No Count ← Count + 1 OUTPUT Count, " positive numbers" END	6

13(b) Any **two** from:

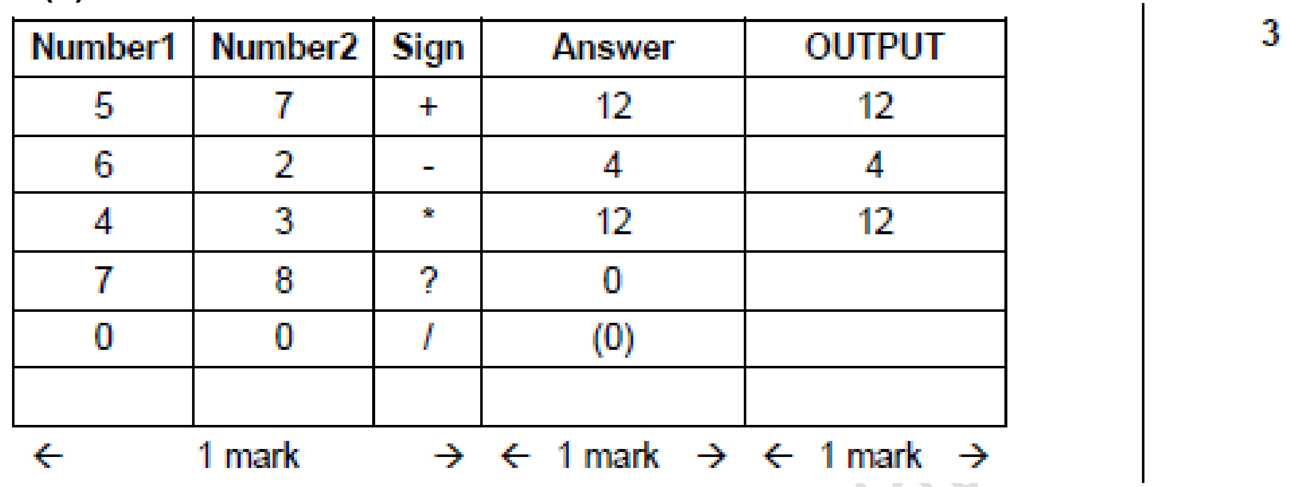
• Use another counter/variable

• Update this counter/variable when the number is less than zero/count all numbers **and** subtract the positive numbers

• Output this counter/variable at the end // Output both counters at the end

[2]

14 (a)



Number1	Number2	Sign	Answer	OUTPUT
5	7	+	12	12
6	2	-	4	4
4	3	*	12	12
7	8	?	0	
0	0	/	(0)	

← 1 mark → ← 1 mark → ← 1 mark →

3

14(b)
CASE Sign OF
□ ENDCASE (1)

List +, -, *, / with correct assignments (1)

OTHERWISE Answer ← 0 (1)

Example CODE

```
CASE Sign OF
'+' : Answer ← Number1 + Number2
'-' : Answer ← Number1 - Number2
'*' : Answer ← Number1 * Number2
'/' : Answer ← Number1 / Number2
OTHERWISE Answer ← 0
ENDCASE
```

[3]

15(a) Max **4** in total
Any **3** from:
- To ensure no changes are made on input / accuracy of transcription
- Because the details do not have fixed, values or lengths to validate
- Because there is no clear set of rules that can be used for validation

Any **3** from:
- The programmer could ask the contributor to type in each detail twice □
- □ and then check that both values are equal
- □ If they are not equal then the input should be rejected
- The programmer could ask the contributor to check the details on the screen
- □ and confirm that they are correct / same as the original
- □ or change them

[4]

15(b) **One** mark for email and **one** mark for password

Email – check for @ / format check / no spaces /valid characters // presence

check // length check (not more than 254 characters) // uniqueness check

Password – length check / numbers and letters etc. // uniqueness check not been used before // presence check

[2]

16
Max **4** in total
Any **3** from:
□ To ensure no changes are made on input / accuracy of transcription
□ Because the details do not have fixed, values or lengths to validate
□ Because there is no clear set of rules that can be used for validation
Any **3** from:
□ The programmer could ask the contributor to type in each detail twice □
□ □ and then check that both values are equal
□ □ If they are not equal then the input should be rejected
□ The programmer could ask the contributor to check the details on the screen □
□ □ and confirm that they are correct / same as the original
□ □ or change them

4

4(b) **One** mark for email and **one** mark for password
Email – check for @ / format check / no spaces /valid characters // presence check // length check (not more than 254 characters) // uniqueness check
Password – length check / numbers and letters etc. // uniqueness check not been used before // presence check

2

17
(a) mark for each error identified + suggested correction
□ `Low ← Count` should be `Low ← Number`
□ `Number > Low` should be `Number < Low`
□ `UNTIL Count = 99` should be `UNTIL Count > 99` or `UNTIL Count = 100` or `UNTIL Count >= 100` // `Count ← 1` should be `Count ← 0`
□ `PRINT "Largest Number is ", Number` should be `PRINT "Largest Number is ", High`

4

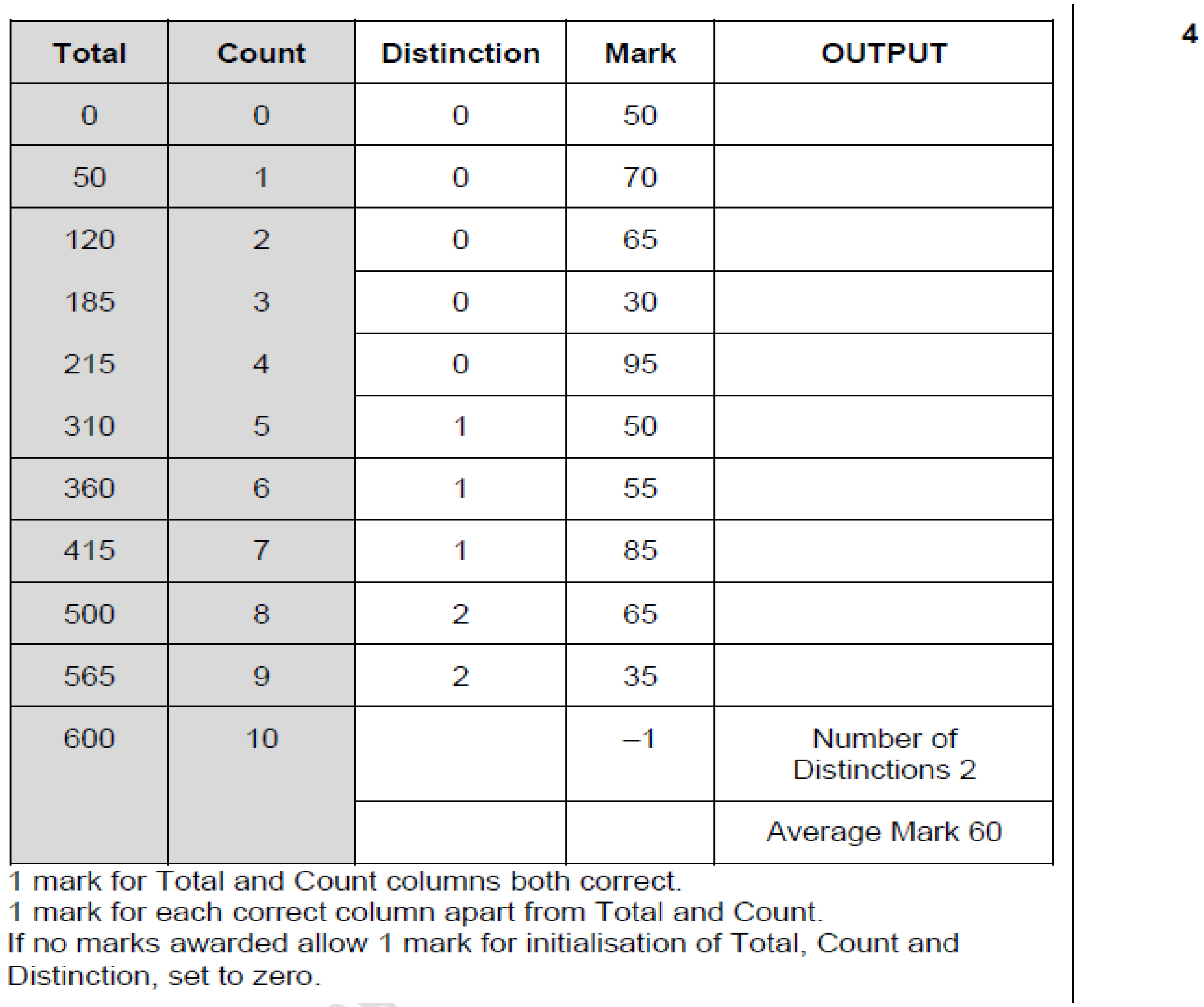
(b) MP1 Add `Total ← 0` // `Total ← Number`
MP2 Add `Total ← Total + Number`
MP3 Add `PRINT "Total is ", Total`

MP4 All positioning explained / seen

```
Count ← 1
INPUT Number
High ← Number
Low ← Number
Total ← Number
REPEAT
INPUT Number
Total ← Total + 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 ″, High
PRINT ″Smallest Number is ″, Low
PRINT ″Total is ″, Total
```

4

18

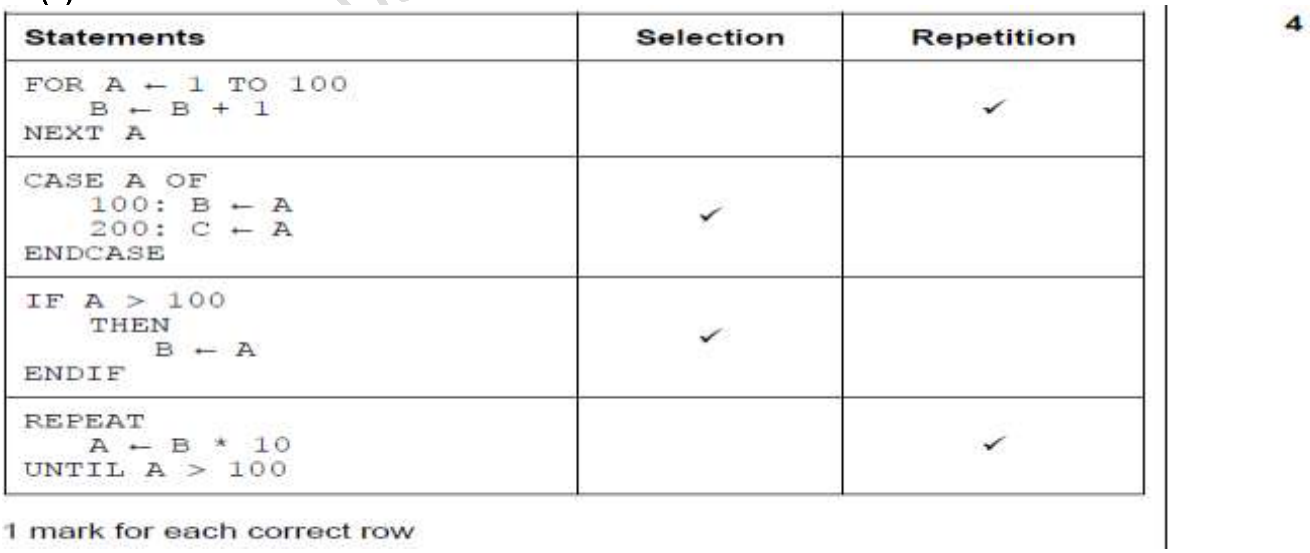


Total	Count	Distinction	Mark	OUTPUT
0	0	0	50	
50	1	0	70	
120	2	0	65	
185	3	0	30	
215	4	0	95	
310	5	1	50	
360	6	1	55	
415	7	1	85	
500	8	2	65	
565	9	2	35	
600	10		–1	Number of Distinctions 2
				Average Mark 60

4

1 mark for Total and Count columns both correct.
1 mark for each correct column apart from Total and Count.
If no marks awarded allow 1 mark for initialisation of Total, Count and Distinction, set to zero.

19 (a)



Statements	Selection	Repetition
FOR A ← 1 TO 100 B ← B + 1 NEXT A		✓
CASE A OF 100: B ← A 200: C ← A ENDCASE	✓	

IF A > 100
THEN
B ← A
ENDIF
	✓	

REPEAT A ← B * 10 UNTIL A > 100		✓

4

1 mark for each correct row

19 (b)

Validation Two from:

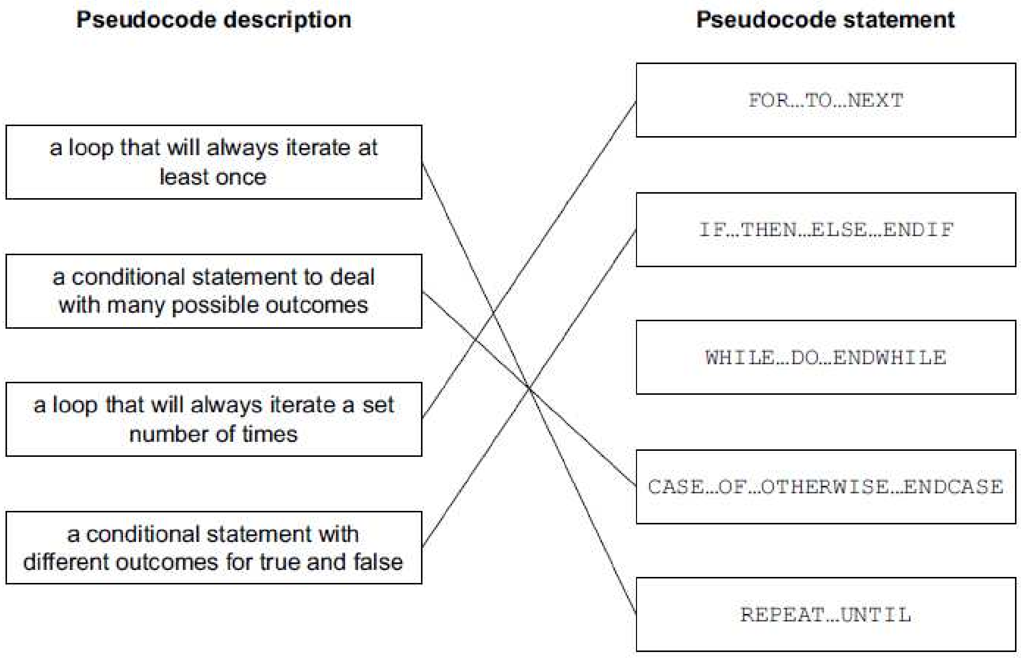
□ automated checking
□ checking that data is reasonable / of a certain type
□ checking that data meets certain criteria Example 1 mark
□ range check // length check // type check // check digit etc.

Verification Two from:

□ checking that data has not changed□
□ □ during input to a computer
□ □ during transfer between computers / devices Example 1 mark
□ double entry // checking against original // visual check // use of checksum etc.

20 (a)

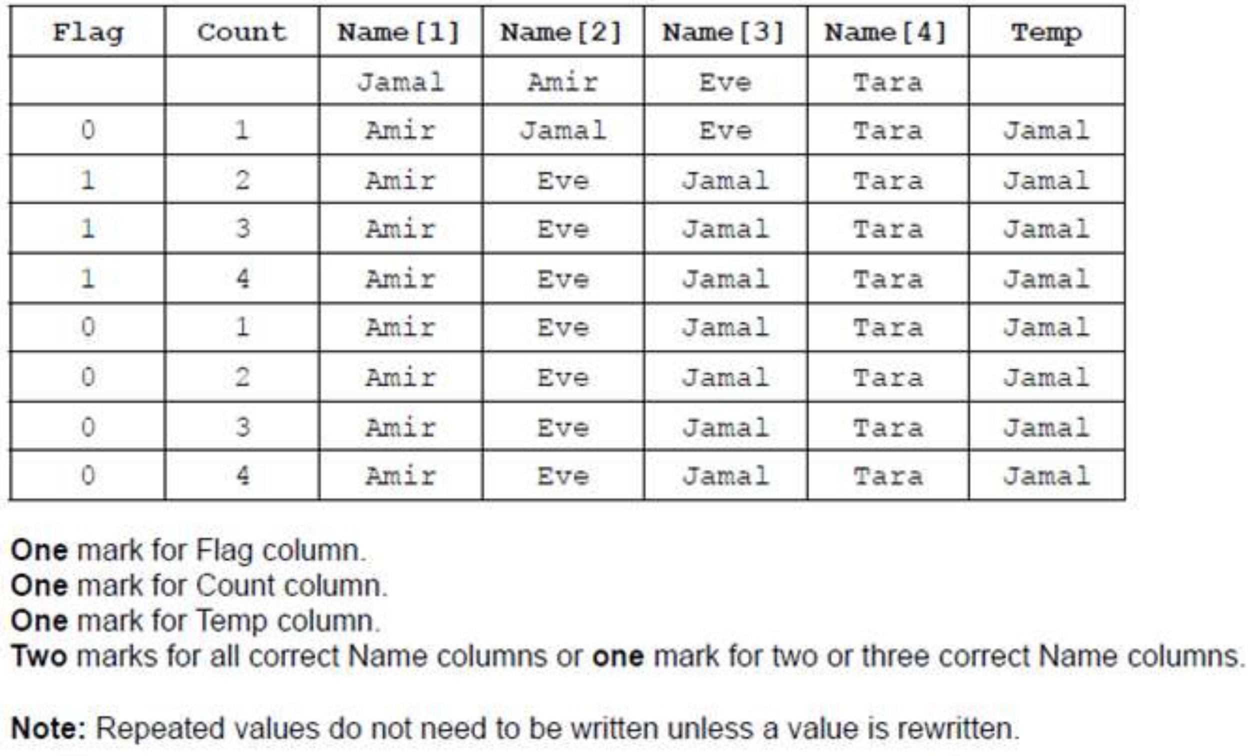
One mark for each correct line.


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

20 (b)

• appropriate loop controls
• read from array
• output from array (the last two points can be in one statement, see example)
 	
Note: reading and the output **MUST** be within the same loop.
 	
For example:
`Count ← 0`
`WHILE Count < 50 DO`
`  OUTPUT Name[Count]`
`  Count ← Count + 1`
`ENDWHILE`
	3

21 (a)



Flag	Count	Name[1]	Name[2]	Name[3]	Name[4]	Temp
		Jamal	Amir	Eve	Tara	
0	1	Amir	Jamal	Eve	Tara	Jamal
1	2	Amir	Eve	Jamal	Tara	Jamal
1	3	Amir	Eve	Jamal	Tara	Jamal
1	4	Amir	Eve	Jamal	Tara	Jamal
0	1	Amir	Eve	Jamal	Tara	Jamal
0	2	Amir	Eve	Jamal	Tara	Jamal
0	3	Amir	Eve	Jamal	Tara	Jamal
0	4	Amir	Eve	Jamal	Tara	Jamal

One mark for Flag column.
One mark for Count column.
One mark for Temp column.
Two marks for all correct Name columns or **one** mark for two or three correct Name columns.

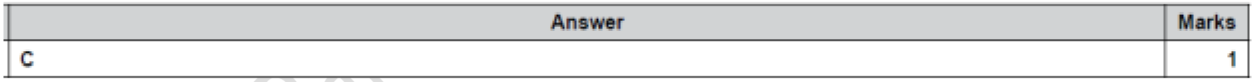
Note: Repeated values do not need to be written unless a value is rewritten.

[5]

21(b)

- bubble sort / sorting the names
- ascending order / A to Z / lowest to highest / alphabetical order [2]

22



Answer	Marks
C	1