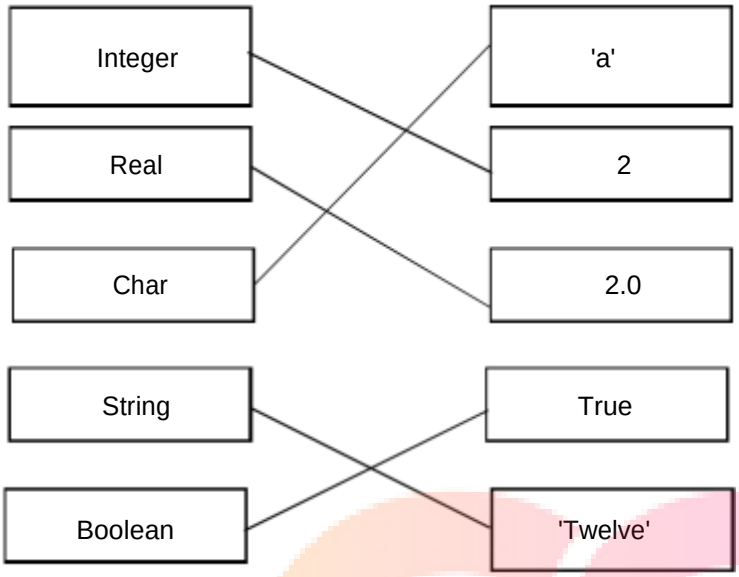
HANDOUT FOR CHAPTERS 10 AND 11

PSEUDOCODE, FLOWCHARTS AND PROGRAMS

Marking Scheme

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



[4]

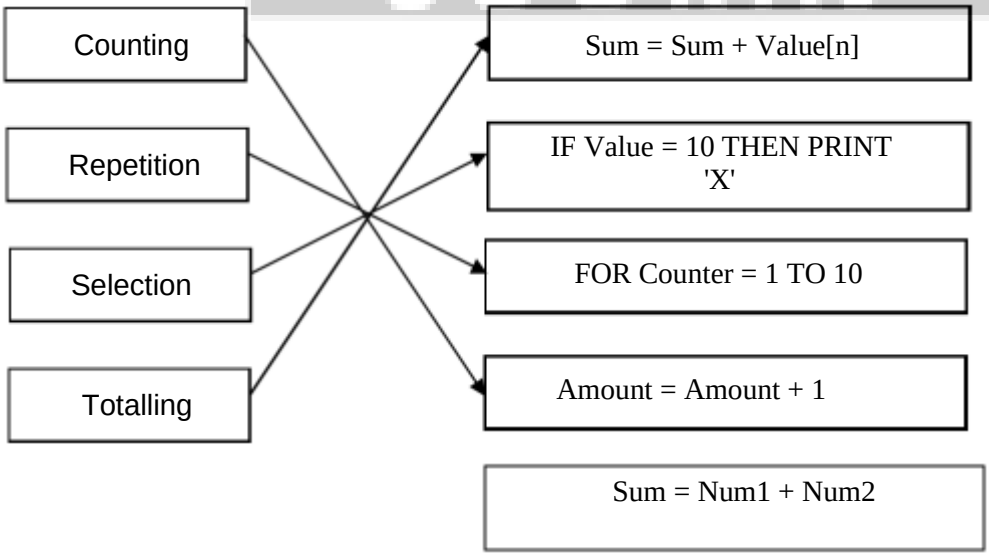
2 Any two points from
- a variable is used to store data that can change during the running of a program
- a constant is used to store data that will not be changed during the running of a program

[2]

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

[3]

4 1 mark for each correct line, two lines from one box not allowed



[4]

5 (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]

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

7 – 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]

8 (i) 1 mark for each improvement

use FOR … NEXT instead of REPEAT … UNTIL
Move PRINT to after the end of the loop
Add error checking to check that the value input is positive [3]

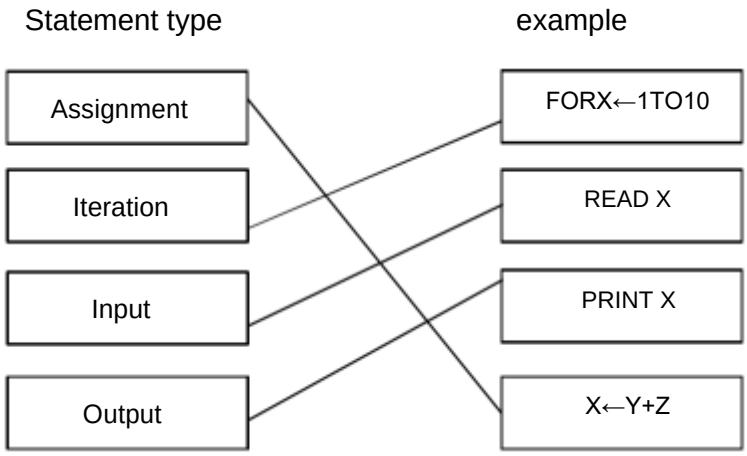
(ii) 3 marks maximum, 1 mark for each improvement correctly included.

Sample answer below

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

[3]

9. 1 mark for each correct line, maximum 3 (zero correct 0, one correct 1, two correct 2, three or four correct 3), each box must have only one connection.



[3]

10 – IF (… THEN … ELSE … ENDIF)

– CASE (… OF … OTHERWISE … ENDCASE) [2]

Question	Answer	Marks

11	-Prompt and input number (1)
	-Checking the input number is between 0 and 100 - both limits (1)
	-Correct error message (1)
	
	Many correct algorithms. This is an example only.
	
	`OUTPUT "Enter a number between 0 and 100 "`
	`INPUT Number`
	`IF Number < 0 OR Number > 100`
	`  THEN`
	`  OUTPUT "The number you have entered is outside the specified range"`
	`ENDIF`
		3

Question	Answer	Marks

12(a)	award full marks for any working solution
	- 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`
	`BigNum1`
	`INPUT Num2, Num3`
	`IF Num2 > Big THEN Big ← Num2 ENDIF`
	`IF Num3 > Big THEN Big ← Num3 ENDIF`
	`PRINT Big`
		4

12(b)	1 mark for each data set and 1 mark for the matching reason. There are many possible correct answers, these are examples only. *Test data set 1:* 30, 29, 28 *Reason:* first number is the largest Test data set 2: x, y, z Reason: abnormal data, should be rejected Max 4 marks	**4**

Question	Answer	Marks

Question	Answer	Marks
13(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**

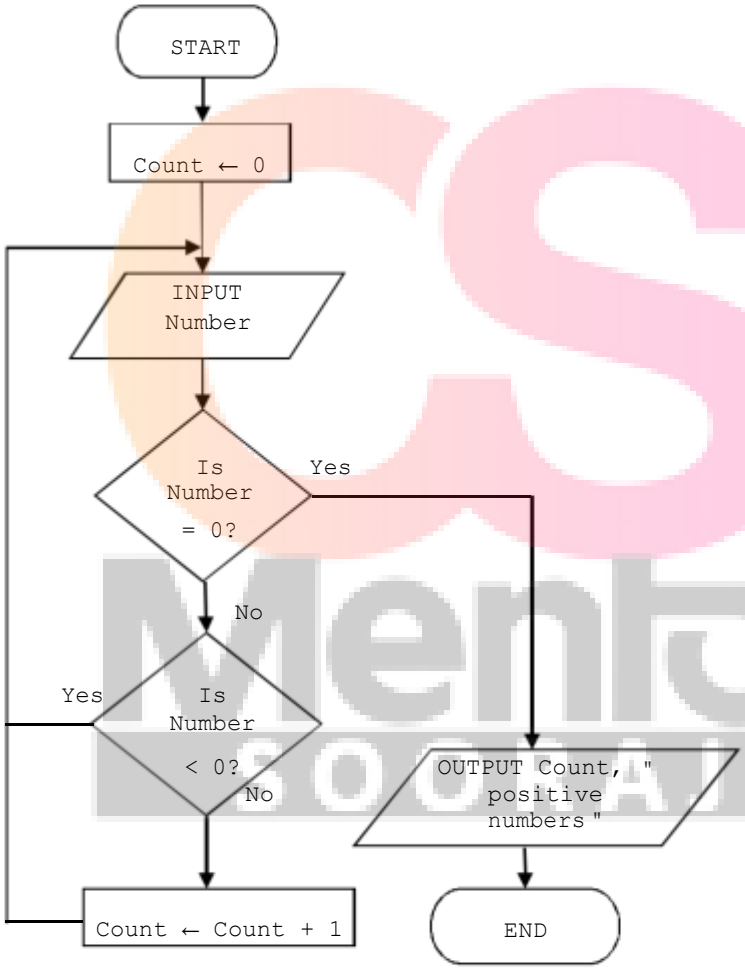
13(b)	- use of `FOR` with correct start and end values
	- use of `NEXT`
	- removal of increment for `Count`
	Sample algorithm
	`Sum0`
	`FOR Count 1 TO 100`
	`INPUT Number`
	`Sum Sum + Number`
	`NEXT // NEXT Count`
	`PRINT Sum`
		3

Question	Answer
	Section B

14(a)	Any **six** from:
	1 Initialisation of counters for positive numbers and zeros
	2 Appropriate loop for 1000 iterations
	3 Input number inside loop
	4 Test for positive numbers
	5 Update positive number counter
	6 Test for zeros
	7 Update zero counter
	8 Output counters with appropriate messages outside loop
	
	`zero ← 0`
	`posCount ← 0`
	`FOR count ← 1 TO 1000`
	`  INPUT number`
	`  IF number > 0`
	`  THEN posCount ← posCount + 1`
	`  ENDIF`
	`  IF number = 0`
	`  THEN zero ← zero + 1`
	`  ENDIF`
	`NEXT`
	`OUTPUT posCount, " positive numbers"`
	`OUTPUT zero, " zeros"`

14(b)	Reduce the number of iterations to a manageable amount // Simulate the input (e.g. random generati

Question	Answer
15	There are many possible answers. e.g.: Totalling is used to sum a list of numbers (1) Counting is used to find how many numbers/items there are in a list. (1) Totalling example (1) e.g. `Total = Total + Number` Counting example (1) e.g. `Counter = Counter + 1`

Question	Answer	Marks
	Section B	
16(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  	**6**
16(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**

Question	Answer	Marks
17(a)	Any **two** from: • Expects a number to be input • Checks if the number is greater than 100 • Outputs the result of the test • Specific output example	**2**
17(b)(i)	**One** mark for correct answer e.g. Use a (condition controlled) loop	**1**

17(b)(ii)	**One** mark for each point
	• Initialisation of Number variable
	• Correct loop statements
	• Correct INPUT and OUTPUT
	e.g.
	`INPUT Number`
	`WHILE Number > 100 DO`
	`  OUTPUT "The number is too large"`
	`  INPUT Number`
	`ENDWHILE`
	`OUTPUT "The number is acceptable"`
	or
	`INPUT Number`
	`REPEAT`
	`  IF Number > 100`
	`  THEN`
	`  OUTPUT "The number is too large"`
	`  ENDIF`
	`  INPUT Number`
	`UNTIL Number <= 100`
	`OUTPUT "The number is acceptable"`
		3

Question	Answer	Marks

Question	Answer	Marks
18	Many possible answers, those given are examples only. 1 mark for each correct description and 1 mark for each correct example **Char** Description: A single character (from the keyboard) Example: A/#/2 **String** Description: An (ordered) sequence of characters Example: Hello world / #123?Y / 234 78963 **Boolean** Description: A data type with two possible values Example: TRUE / FALSE	**6**

Question	Answer	Marks

19(a)	Many possible answers, those given are examples only.
	1 mark per bullet:
	• IF
	• Condition and outcome
	
	Example answer:
	`IFX<0`
	`  THEN`
	`  PRINT "Negative"`
	`  ELSE`
	`  PRINT "Not negative"`
	`ENDIF`
	
	OR
	
	1 mark per bullet:
	• CASE
	• Condition and outcome
	
	Example answer:
	`CASE X OF`
	`  1: PRINT ("ONE")`
	`  2: PRINT ("TWO")`
	`  OTHERWISE PRINT ("Less than ONE or more than TWO")`
	`ENDCASE`
		2

19(b)	• To allow different routes through a program • dependent on meeting certain criteria	**2**

Question	Answer	Marks
20		4

Statements	Selection	Repetition
FORA←1TO100 B←B+1 NEXT A		✓
CASE A OF 100:B←A 200:C←A ENDCASE	✓	
IFA>100 THEN B ← A ENDIF	✓	
REPEAT A←B*10 UNTIL A > 100		✓

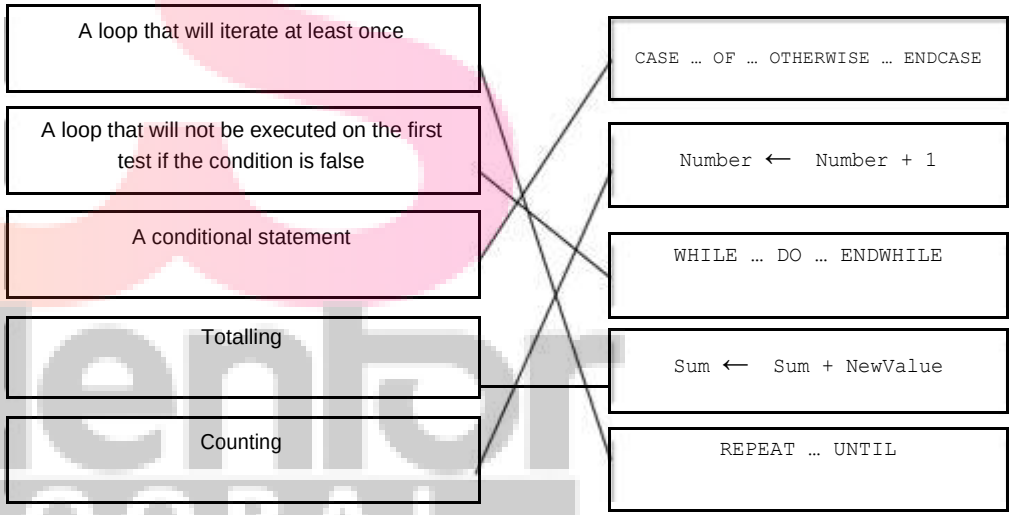
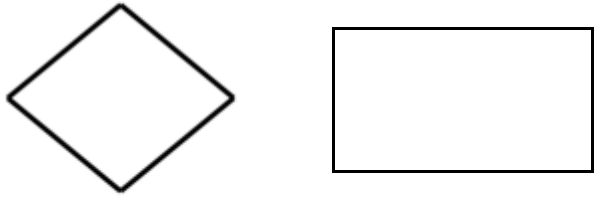
1 mark for each correct row

Question	Answer	Marks
21	☐ FOR … TO … NEXT ☐ fixed number of repetitions ☐ REPEAT … UNTIL ☐ always executed // condition tested at end ☐ WHILE … DO … ENDWHILE ☐ may not be executed // condition tested at beginning	6

Question	Answer	Marks
22(a)	Any **two** correct statements (max **two**) e.g. • The value of the variable Count begins as 0 … • … and is incremented by 1 before it is tested by the loop condition • Count will never be 0 at the end of the loop	**2**

Question	Answer	Marks
22(b)	``` Count ← 0 REPEAT INPUT Number IF Number >= 100 THEN Values[Count] ← Number ENDIF Count ← Count + 1 UNTIL Count = 50 ``` **One** mark – separate INPUT statement **One** mark – IF statement attempted **One** mark – IF statement completely correct **One** mark – termination of loop updated	**4**
22(c)	Any **two** correct statements (max **two**) e.g. • Alter the IFstatement/add a second IF statement/comparison that's already there … • … so that additional criteria set an upper limit of <=200	**2**

Question	Answer	Marks
23	Real Integer Char/String String Boolean	**5**

24	**Description** / **Pseudocode example**  **One** mark – **one** correct link **Two** marks – **two** correct links **Three** marks – **three** correct links **Four** marks – **four/five** correct links	**4**
25	 **One** mark for each correct symbol	