Loops 4 MS

Question	Answer	Marks

1	**Correct code example:**
	`DECLARE Mark, Sum, Average : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Sum <- 0`
	
	1 mark for correct loop counter
	`//Iterate 10 times`
	`FOR i <- 1 TO 10`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter a mark"`
	`INPUT Mark`
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Mark < 0 OR Mark > 40 DO`
	`OUTPUT "Invalid mark, try again"`
	`INPUT Mark`
	`ENDWHILE`
		7

	1 mark for correct calculation
	`Sum <- Sum + Mark`
	`NEXT`
	
	1 mark for correct calculation and output
	`//Calculate and output results`
	`Average <- Sum / 10`
	`OUTPUT "Average is ", Average`
	1 mark for appropriate comments	

Question	Answer	Marks

2	**Correct code example:**
	`DECLARE Temp, Sum, Average : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Sum <- 0`
	
	1 mark for correct loop counter
	`//Iterate 30 times`
	`FOR i <- 1 TO 30`
		7

	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter today's temperature"`
	`INPUT Temp`
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Temp < -19 OR Temp > 40 DO`
	`  OUTPUT "Invalid temperature, try again"`
	`  INPUT Temp`
	`ENDWHILE`
	
	1 mark for correct calculation
	`Sum <- Sum + Temp`
	`NEXT`
	
	1 mark for correct calculation and output
	`//Calculate and output results`
	`Average <- Sum / 30`
	`OUTPUT "Average is ", Average`
	1 mark for appropriate comments	

Question	Answer	Marks

3	**Correct code example:**
	`DECLARE Num, Min : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Min <- 99999`
	1 mark for correct loop counter
	`//Iterate 100 times`
	`FOR i <- 1 TO 100`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter a number"`
	`INPUT Num`
	1 mark for correct condition and assignment
	`IF Num < Min THEN`
	`Min <- Num`
	`ENDIF`
	`NEXT`
	1 mark for correct output
	`//Output results`
	`OUTPUT "Minimum is ", Min`
		5

Question	Answer	Marks

4	**Correct code example:**
	`DECLARE Mark : INTEGER`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter a mark"`
	`INPUT Mark`
	
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Mark < 0 OR Mark > 100 DO`
	`    OUTPUT "Invalid mark, try again"`
	`    INPUT Mark`
	`ENDWHILE`
	
	1 mark for correct condition and outputs
	`//Output results`
	`IF Mark >= 50 THEN`
	`    OUTPUT "Passed"`
	`ELSE`
	`    OUTPUT "Failed"`
	`ENDIF`
		3

Question	Answer	Marks

5	**Correct code example:**
	`DECLARE Temp, Max : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Max <- -21    //Any number less than -20 works fine`
	
	1 mark for correct loop counter
	`//Iterate 365 times`
	`FOR i <- 1 TO 365`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter today's temperature"`
	`INPUT Temp`
	
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Temp < -20 OR Temp > 100 DO`
	`OUTPUT "Invalid temperature, try again"`
	`INPUT Temp`
	`ENDWHILE`
		7

	1 mark for correct condition and assignment
	`IF Temp > Max THEN`
	`  Max <- Temp`
	`ENDIF`
	`NEXT`
	
	1 mark for correct outputs
	`//Output results`
	`OUTPUT "Maximum is ", Max`
	1 mark for appropriate comments	

Question	Answer	Marks

6	**Correct code example:**
	`DECLARE Mark, Sum, Average : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Sum <- 0`
	
	1 mark for correct loop counter
	`//Iterate 50 times`
	`FOR i <- 1 TO 50`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter a mark"`
	`INPUT Mark`
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Mark < 0 OR Mark > 100 DO`
	`OUTPUT "Invalid mark, try again"`
	`INPUT Mark`
	`ENDWHILE`
	1 mark for correct calculation
	`Sum <- Sum + Mark`
	`NEXT`
		8

	1 mark for correct calculation
	`Average <- Sum / 50`
	1 mark for correct condition and outputs
	`//Output results`
	`IF Average >= 70 THEN`
	`    OUTPUT "Good Performance"`
	`ELSE`
	`    OUTPUT "Bad Performance"`
	`ENDIF`
	1 mark for appropriate comments	

Question	Answer	Marks

7	**Correct code example:**
	`DECLARE Mark, Sum, Average : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Sum <- 0`
	1 mark for correct loop counter
	`//Iterate over the number of students`
	`FOR i <- 1 TO StudentCount`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter a mark"`
	`INPUT Mark`
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Mark < 0 OR Mark > 40 DO`
	`OUTPUT "Invalid mark, try again"`
	`INPUT Mark`
	`ENDWHILE`
	1 mark for correct calculation
	`Sum <- Sum + Mark`
	`NEXT`
		7

	1 mark for correct calculation and output
	`//Calculate and output results`
	`Average <- Sum / StudentCount`
	`OUTPUT "Average is ", Average`
	1 mark for appropriate comments	

Question	Answer	Marks
8	**Error 1:** Line 1 OR `Large <- 9999` **Correction:** `Large <- -9999` **Error 2:** `Line 3  OR  WHILE Counter > 30` **Correction:** `WHILE Counter < 30` **Error 3:** Line 6 OR `IF Num < Large` **Correction:** `IF Num > Large`	**3**

Question	Answer	Marks
9	**Error 1:** Line 1 OR `Small <- 0` **Correction:** `Small <- 9999` **Error 2:** `Line 5  OR  Num <- Small` **Correction:** `Small <- Num` **Error 3:** Line 7 OR `OUTPUT Small` **Correction:** Should be after `UNTIL // Line 8` **Error 4:** `Line 8  OR UNTIL Counter < 10` **Correction:** `UNTIL Counter = 10`	**4**

Question	Answer	Marks

10	**Correct code example:**
	`DECLARE Mark, Max : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Max <- -1    //Any number less than 0 works fine`
	
	1 mark for correct loop counter
	`//Iterate 100 times`
	`FOR i <- 1 TO 100`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter a mark"`
	`INPUT Mark`
	
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Mark < 0 OR Mark > 100 DO`
	`OUTPUT "Invalid mark, try again"`
	`INPUT Mark`
	`ENDWHILE`
		6

	1 mark for correct condition and assignment
	`IF Mark > Max THEN`
	`  Max <- Mark`
	`ENDIF`
	`NEXT`
	
	1 mark for correct output
	`//Output results`
	`OUTPUT "Maximum is ", Max`	

Question	Answer	Marks

11	**Correct code example:**
	`DECLARE Age, Allowed, Rejected : INTEGER`
	1 mark for initialisation
	`//Initialisation`
	`Allowed <- 0`
	`Rejected <- 0`
	
	1 mark for correct loop counter
	`//Iterate 150 times`
	`FOR i <- 1 TO 150`
	1 mark for correct input with appropriate prompt message
	`OUTPUT "Enter an age"`
	`INPUT Age`
	
	1 mark for correct validation using WHILE or REPEAT
	`WHILE Age < 1 OR Age > 80 DO`
	`OUTPUT "Invalid age, try again"`
	`INPUT Age`
	`ENDWHILE`
		6

	1 mark for correct condition and calculations
	`IF Age > 12 THEN`
	`  Allowed <- Allowed + 1`
	`ELSE`
	`  Rejected <- Rejected + 1`
	`ENDIF`
	`NEXT`
	
	1 mark for correct outputs
	`//Output results`
	`OUTPUT "Allowed players: ", Allowed`
	`OUTPUT "Rejected players: ", Rejected`	

Question	Answer	Marks

12	**Correct code example:**
	`DECLARE Mark, Sum, Average, Maximum : REAL`
	`DECLARE Count : INTEGER`
	1 mark for initialisation
	`//Initialisation`
	`Sum <- 0`
	`Count <- 0`
	`Maximum <- -999999`
	1 mark for correct loop counter
	`//Iterate over the number of students`
	`FOR i <- 1 TO StudentCount`
	`    OUTPUT "Enter a mark"`
	`    INPUT Mark`
	1 mark for correct input with appropriate prompt message and validation using WHILE or REPEAT
	`    WHILE Mark < 0 OR Mark > 40 DO`
	`        OUTPUT "Invalid mark, try again"`
	`        INPUT Mark`
	`    ENDWHILE`
		8

1 mark for correct calculation

```
    IF Mark > 30 THEN
        Sum <- Sum + Mark
        Count <- Count + 1
    ENDIF
```

1 mark for correct maximum condition

```
    IF Mark > Maximum THEN
        Maximum <- Mark
    ENDIF
NEXT
```

1 mark for correct calculation and output

```
//Calculate and output results
IF Count > 0 THEN
    Average <- Sum / Count
    OUTPUT "Average is ", Average
ELSE
    OUTPUT "No Average"
ENDIF
```

	1 mark for correct conditions and outputs
	`OUTPUT "Maximum is ", Maximum`
	`IF Maximum >= 35 THEN`
	`    OUTPUT "Good"`
	`ELSE`
	`    IF Maximum >= 30 THEN`
	`        OUTPUT "Moderate"`
	`    ELSE`
	`        OUTPUT "Weak"`
	`    ENDIF`
	`ENDIF`
	1 mark for appropriate comments	

Question	Answer	Marks

13	**Correct code example:**
	1 mark for correct loop counter
	`//Iterate 10 times`
	`FOR i <- 1 TO 10`
	`    OUTPUT i`
	`NEXT`
	1 mark for correct output
	1 mark for appropriate comments
		3

Question	Answer	Marks

14	**Correct code example:**
	1 mark for correct loop count and step
	`//Iterate 10 times`
	`FOR i <- 10 TO 1 STEP -1`
	`    OUTPUT i`
	`NEXT`
	1 mark for correct output
	1 mark for appropriate comments
		3

Question	Answer	Marks

15	**Correct code example:**
	OPTION 1
	`DECLARE Count : INTEGER`
	1 mark for correct condition and initialisation
	`Count <- 0`
	`WHILE Count < 5 DO` `Condition can also be Count <= 4`
	1 mark for correct output
	`OUTPUT "Hello World"`
	1 mark for correct incrementing of count
	`Count <- Count + 1`
	`ENDWHILE`
		3

	OPTION 2 DECLARE Count : INTEGER **1 mark for correct condition and initialisation** Count <– 1 WHILE Count <= 5 DO Condition can also be Count < 6 **1 mark for correct output** OUTPUT “Hello World” **1 mark for correct incrementing of count** Count <– Count + 1 ENDWHILE	

Question	Answer	Marks

16	**Correct code example:**
	OPTION 1
	`DECLARE Count : INTEGER`
	1 mark for correct condition and initialisation
	`Count <- 0`
	`Repeat`
	1 mark for correct output
	`OUTPUT "Hello World"`
	1 mark for correct incrementing of count
	`Count <- Count + 1`
	`UNTIL Count = 5` `Condition can also be Count > 4 or Count >= 5`
		3

	OPTION 2 DECLARE Count : INTEGER **1 mark for correct condition and initialisation** Count <– 1 Repeat **1 mark for correct output** OUTPUT “Hello World” **1 mark for correct incrementing of count** Count <– Count + 1 UNTIL Count = 6 Condition can also be Count > 5 or Count >= 6	

Question	Answer	Marks

17	**Correct code example:**
	`DECLARE Salary, Sum, Average, Maximum, Count : REAL`
	1 mark for initialisation
	`//Initialisation`
	`Sum <- 0`
	`Count <- 0`
	`Maximum <- -999999`
	1 mark for correct loop counter
	`//Iterate over the number of employees`
	`FOR i <- 1 TO FirmSize`
	`OUTPUT "Enter a salary"`
	`INPUT Salary`
	1 mark for correct input with appropriate prompt message and validation using WHILE or REPEAT
	`WHILE Salary < 300 OR Salary > 60000 DO`
	`OUTPUT "Invalid salary, try again"`
	`INPUT Salary`
	`ENDWHILE`
		8

1 mark for correct calculation

```
    Sum <- Sum + Salary
```

1 mark for correct maximum condition

```
    IF Salary > Maximum THEN
        Maximum <- Salary
    ENDIF
```

1 mark for correct condition and count calculation

```
    IF Salary > 30000 THEN
        Count <- Count + 1
    ENDIF
NEXT
```

1 mark for correct calculation and output

```
//Calculate and output results
Average <- Sum / FirmSize
OUTPUT "Average is ", Average
```

1 mark for correct outputs

```
OUTPUT “Maximum is “, Maximum
OUTPUT “Count is “, Count
```

Question	Answer	Marks
18(a)	**Assignment statement:** Line 1 and Line 2 and Line 6 and Line 10 **Selection statement:** Line 4 and Line 8 **Iteration statement:** Line 3	**3**
18(b)	**Error 1:** Line 4 **Correction:** IF List[Counter] **>** Max **Error 2:** Line 8 **Correction:** IF List[**Counter**] < Min **Error 3:** Line 11 **Correction:** Should be ENDIF	**3**