

Past Papers May/June 2015 to 2018:

9608/21/22/MJ/15

Q2 A program displays a menu with choices 1 to 4. The code to display the menu is written as the procedure DisplayMenu.

(a) Pseudocode which uses this procedure is

```

CALL DisplayMenu
REPEAT
  OUTPUT "Enter choice (1..4)"
  INPUT Choice
UNTIL Choice >= 1 AND Choice <= 4

```

(i) Describe what this pseudocode will do.

.....

.....

.....

.....[3]

(ii) State why a loop is required.

.....

.....[1]

(b) The following pseudocode is a revised design.

```

CONSTANT i ← 3
CALL DisplayMenu
NoOfAttempts ← 0
REPEAT      OUTPUT "Enter choice (1..4)"
  INPUT Choice
  NoOfAttempts ← NoOfAttempts + 1
UNTIL (Choice >= 1 AND Choice <= 4) OR NoOfAttempts = i

```

(i) Give the maximum number of inputs the user could be prompted to make.

..... [1]

(ii) State why this algorithm is an improvement on the one given in part (a).

.....

.....[1]

(c) The pseudocode is in its initial stage of development.

The table below shows the action currently taken by the pseudocode following each menu choice.

Menu choice	Description	Program response
1	Read data from the customer file	Calls a procedure ReadFile which for testing purposes outputs the message "Read file code"
2	Add a customer	Outputs message "Add customer code"
3	Search for a customer	Outputs message "Search customer code"
4	Terminates the program	Ends



Answer

9608/21/22/MJ/15

2 (a) (i) Displays the menu (choices) Repeats the prompt and inputthe input is a number between 1 and 4 // Checks number is between 1 and 4

"within range" is not enough [3]

(ii) ...the input number is validated [1]

(b) (i) 3 [1]

(ii) Previous design repeated indefinitely // (new design) limits number of attempts

Penalise "Program terminates/closes" [1]

(c) IF Choice = 1 THEN (CALL) ReadFile (1)
IF Choice = 2 THEN OUTPUT "Add Customer code" (1)
IF Choice = 3 THEN OUTPUT "Search Customer code" (1)
IF Choice = 4 THEN END (1)

alternative answer:

mark as follows:

CASE OF Choice // Select CASE Choice 1 mark

1: (CALL) ReadFile 1 mark (allow CASE = 1)

2: OUTPUT "Add Customer code" 1 mark


3: OUTPUT "Search Customer code" 1 mark


4: END ENDCASE


Output strings must match

[max 3]


(d) Mark as follows:


 Choice / NoOfAttempts declared/commented as integer Must appear within the 'main' program Allow: different identifier names


 Constant i assigned a value 3

 There is an 'outer' loop to repeatedly display the menu


 Input 'choice' variable


 Three IF statements (or equivalent) for processing menu choices 1, 2 and 3 Note: they must be correctly formed as 'nested' or 'independent'

 Choice 1 calls procedure ReadFile

 Choice 2 outputs "Add Customer Code" + Choice 3 outputs "Search Customer Code"

 Outer loop terminates correctly with 'Choice = 4' //or equivalent

 Procedure DisplayMenu shows the four menu options

 Procedure ReadFile is present ... and contains a single output message 'Read file code'
[max 8]

(d) Answer
Sample VB Program

```

Module Module1
    Const i As Integer = 3
    Dim NoOfAttemp As Integer
    Dim choice As Integer
    Sub DisplayMenu()
        Console.WriteLine("Options are ")
        Console.WriteLine("1 = READ DATA FROM THE CUSTOMER FILE")
        Console.WriteLine("2 = Add a Customer")
        Console.WriteLine("3 = Search for Customer")
        Console.WriteLine("4 = Terminate the Program")
    End Sub

    Sub ReadFile()
        Console.WriteLine("Read Customer File Record")
    End Sub
    Sub AddCust()
        Console.WriteLine("Addition of Customer Done")
    End Sub
    Sub searchCust()
        Console.WriteLine("Search Customer done")
    End Sub

    Sub Main()
        Do
            DisplayMenu()
            Console.WriteLine("Enter your Choice 1 to 4")
            choice = Console.ReadLine()
            If choice = 1 Then
                ReadFile()
            ElseIf choice = 2 Then
                AddCust()
            ElseIf choice = 3 Then
                searchCust()
            ElseIf choice = 4 Then
                GoTo 1
            End If
            NoOfAttemp = NoOfAttemp + 1
        Loop Until choice >= 1 And choice <= 4 And NoOfAttemp = i
    End Sub
End Module

```

9608/21/22/MJ/16

4 (a) Structured programming involves the breaking down of a problem into modules.
Give two reasons why this is done.

1

2

[2]



9608/21/22/MJ/16

Answer

4	(a)	<ul style="list-style-type: none"> • Program code is <u>easier</u> to implement / manage • Modules may be given to different people to develop // given to program specialists • Program code is <u>easier</u> to test / debug / maintain • Encourages the re-usability of program code 	Max 2
---	-----	---	--------------

9608/23/MJ/17

5 A multi-user computer system records user login information in a text file, LoginFile.txt. Each time a user successfully logs into the system, the following information is recorded:

Item	Information	Example data
1	A five character user ID	"JimAA"
2	A four character port ID	"3456"
3	A fourteen character time and date	"08:30Jun012015"

The data items are concatenated to form a single string. Each string is saved as a separate line in the text file.

The example data in the preceding table would result in the following text line in the file:

"JimAA345608:30Jun012015"




The computer system can produce a list of the successful login attempts by a given user.

The file **LoginFile.txt** is searched for a given user ID and the corresponding data are copied into a 2D array, **LoginEvents**.

LoginEvents has been declared in pseudocode as:

DECLARE LoginEvents[1 : 1000, 1 : 2] OF STRING

A procedure, **SearchFile**, is needed to search the file and copy selected data to the array. The main steps of the procedure are as follows:

-  Input a user ID.
-  Search **LoginFile.txt** for entries with matching user ID.
-  For matching entries, copy items 2 and 3 above into the **LoginEvents** array.

You can assume that:

Programming Code Example Solutions

Q5 : Visual Basic

```
Sub SearchFile()  
    Dim FileData As String  
    Dim SearchID As String  
    Dim ArrayIndex As Integer  
  
    ArrayIndex = 1  
    FileOpen(1, "LoginFile.txt", OpenMode.Input)  
    SearchID = Console.ReadLine()  
  
    Do While Not EOF(1)  
        FileData = LineInput(1)  
        If SearchID = LEFT(FileData, 5) Then  
            LoginEvents(ArrayIndex, 1) = Mid(Filedata, 6, 4)  
            LoginEvents(ArrayIndex, 2) = Right(Filedata, 14)  
            ArrayIndex = ArrayIndex + 1  
        End If  
    Loop  
    FileClose(1)  
End Sub
```

Alternative:

```
Sub SearchFile()  
    Dim FileData As String  
    Dim SearchID As String  
    Dim ArrayIndex As Integer  
    Dim MyFile As System.IO.StreamReader  
  
    ArrayIndex = 1  
    MyFile = Mycomputer.FileSystem.OpenTextFileReader("Loginfile.txt")  
    SearchID = Console.ReadLine()  
  
    Do While MyFile.Peek < > -1  
        FileData = MyFile.ReadLine()  
        If SearchID = LEFT(FileData, 5) Then  
            LoginEvents(ArrayIndex, 1) = Mid(Filedata, 6, 4)  
            LoginEvents(ArrayIndex, 2) = Right(Filedata, 14)  
            ArrayIndex = ArrayIndex + 1  
        End If  
    Loop  
    MyFile.Close  
End Sub
```

9608/22/MJ/18

3 A chocolate factory produces bars of chocolate. A computer program controls the process.

The weight of each bar is stored in an array, **BarWeight**. The array contains 100 elements, representing the weights of 100 bars that make up one shipping box.



A procedure, **CheckWeight()**, is required to:

1. examine each array element and count how many times the weight has exceeded **MaxWeight**
2. compare the count obtained with a limit value, **Threshold**. Call procedure **ServiceCheck()** if the count exceeds the **Threshold**
3. output a message if the count does not exceed the **Threshold**. For example:

"ShippingBox OK – maximum weight exceeded 3 times."

Draw a program flowchart on the next page to represent the algorithm for the **CheckWeight()** procedure.

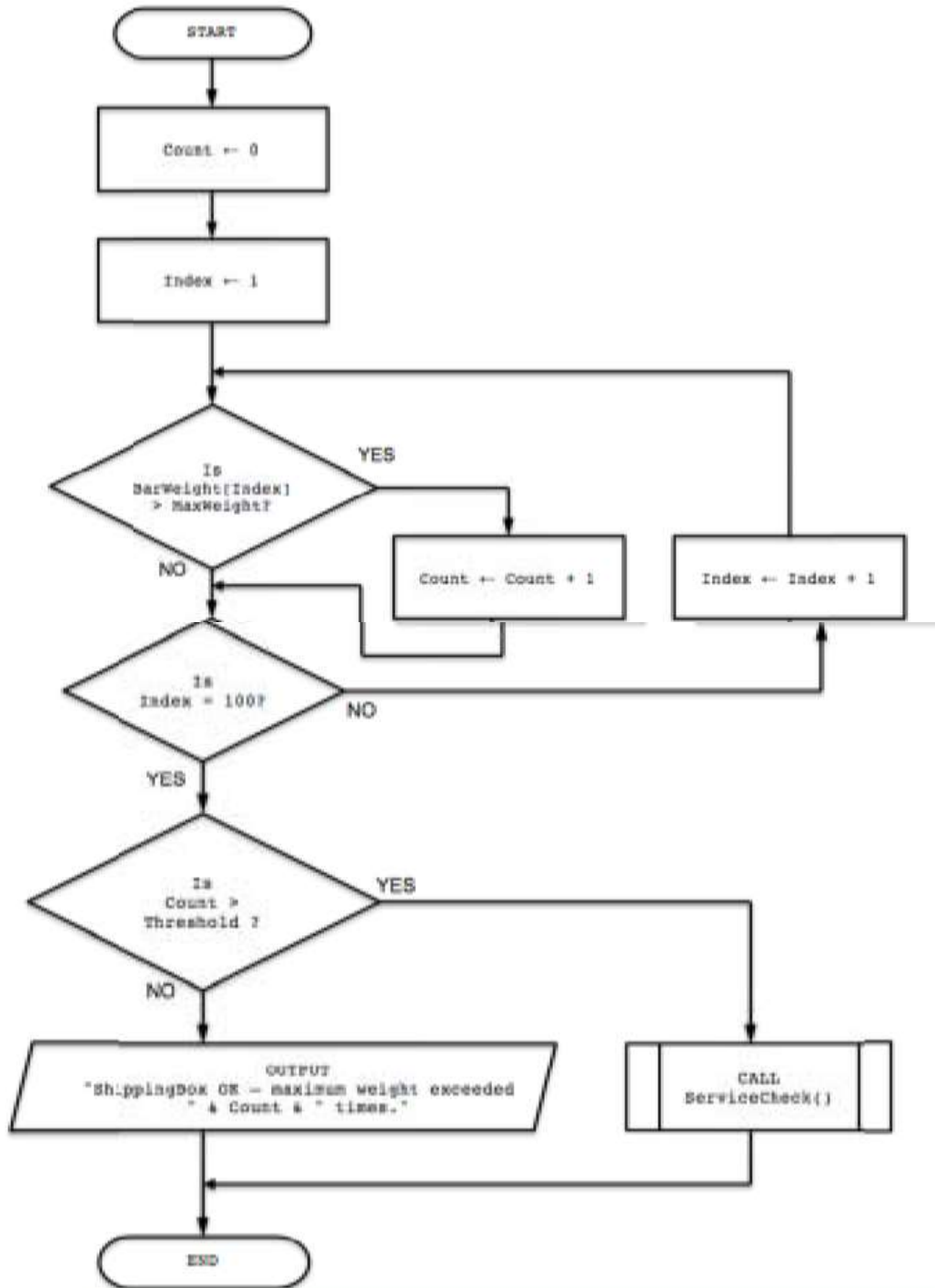
Assume that:

-  the array contains 100 valid weight values and the first element is **BarWeight[1]**
-  **MaxWeight**, **Threshold** and **BarWeight** are global variables. Variable declarations are not required in program flowcharts.

9608/22/MJ/18

Answer

3 Example Program Flowchart



9608/21/ON/15

Q.6

The program design in pseudocode is produced as follows:

```
01 DECLARE StaffNum : INTEGER
02 DECLARE TaskNum : INTEGER
03 DECLARE
.....
.....
04 DECLARE NewStaffTask : BOOLEAN
05
06 CALL InitialiseTaskGrid
07 Completed ← 0
08 WHILE Completed <> 60
09     NewStaffTask ← FALSE
10     WHILE NewStaffTask = FALSE
11         StaffNum ← RANDOM(1,5) //generates a random number
12         TaskNum ← RANDOM(1,12) //in the given range
13         IF TaskGrid[StaffNum, TaskNum] = FALSE
14             THEN
15                 TaskGrid[StaffNum, TaskNum] ← TRUE
16                 NewStaffTask ← TRUE
17                 OUTPUT StaffNum, TaskNum
18             ENDIF
19     ENDWHILE
20     Completed ← Completed + 1
21 ENDWHILE
22 OUTPUT "Staff Task Count", Completed
23
24 // end of main program
25
26 PROCEDURE InitialiseTaskGrid()
27     DECLARE i : INTEGER
28     DECLARE j : INTEGER
29     FOR i ← 1 TO 5
30         FOR j ← 1 TO 12
31             TaskGrid[i, j] ← FALSE
32         ENDFOR
33     ENDFOR
34 ENDPROCEDURE
```

Study the pseudocode and answer the questions below.

Give the line number for:

- (i) The declaration of a BOOLEAN global variable. [1]
- (ii) The declaration of a local variable. [1]



- (iii) The incrementing of a variable used as a counter, but not to control a 'count controlled' loop. [1]
- (iv) A statement which uses a built-in function of the programming language. [1]

(c)

- (i) State the number of parameters of the InitialiseTaskGrid procedure. [1]
- (ii) Copy the condition which is used to control a 'pre-condition' loop.[1]
- (iii) Explain the purpose of lines 13 – 18.
.....
.....
.....
.....
.....[3]
- (iv) Give the global variable that needs to be declared at line 03.
.....[2]

9608/21/ON/15

Answer

- 6 (a) Combination of staff and task number // the pair of numbers // the pair of random numbers [1]
//there will be duplicates /repeats//some staff tasks will not be generated [1]
- (b) (i) 04 // 03 [1]
- (ii) 27 // 28 [1]
- (iii) 20 [1]
- (iv) 11 / 12 [1]
- (c) (i) Zero [1]
- (ii) Completed <> 60 // NewStaffTask = FALSE [1]
Allow: Inclusion of the WHILE
- (iii) Determines whether this combination of StaffNum and TaskNum has been completed [1]
Assigns value TRUE if not already generated [1]
Flags that this is the first time this staff + task has been selected/to exit the loop [1]
Outputs the new staff + task number [1]
- [MAX 3]
- (iv) TaskGrid : ARRAY[1:5, 1:12] OF BOOLEAN
1 mark | 1 mark [2]

