Chapter 11 Algorithm Design and Problem Solving

Chapter 13 Programming and Data Representation: Answers to coursebook questions and tasks

Syllabus sections covered: 2.1 (2.1.1), 2.2, 2.3 (2.3.1 – 2.3.5), 2.4 (2.4.1)

It is suggested that Chapter 11 is worked through in parallel with Chapter 13.

Task 11.01

The following are examples of answers. These examples show the sort of detail students should show in their answers.

To make a sandwich	To walk from college to shop	To log on to computer
Cut two slices of bread Spread butter on one side of each slice of bread Lay a slice of cheese on the buttered side of one slice of bread Cover the cheese with the second slice of buttered bread, the buttered side facing the cheese.	Exit college through the main entrance Turn right and walk to T-junction Turn left and walk 50 metres Cross the road: you now stand in front of the shop.	Power up computer Wait for log-on screen Enter username in first text box Enter password in second text box Press Enter.

Task 11.02

Identifier	Explanation
Inches	Length as a whole number of inches
Cm	The result from using the given formula: Cm = Inches * 2.54

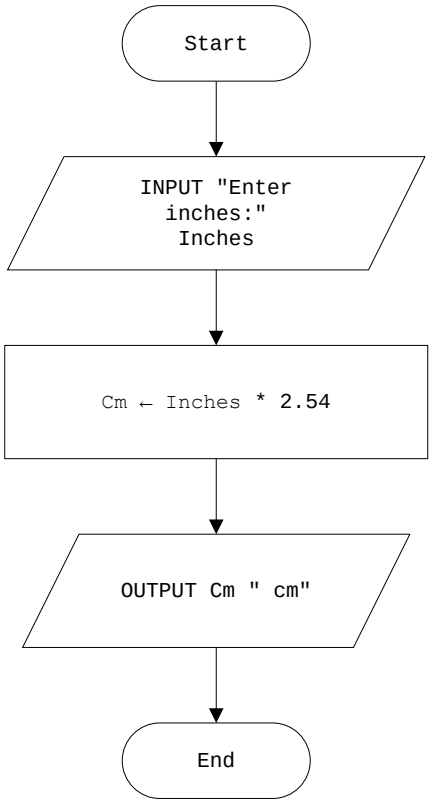

Figure 11.6

```
INPUT "Enter inches: " Inches
Cm ← Inches * 2.54
OUTPUT Cm " cm"
```

Task 11.03

IF Age < 12 OR Age > 60 THEN fare is free

Question 11.01

```
IF Number1 < Number2
   THEN // Number1 is smaller
      IF Number1 < Number3
         THEN
            OUTPUT Number1
         ELSE
            OUTPUT Number3
      ENDIF
ELSE // Number2 is smaller
      IF Number2 < Number3
         THEN
            OUTPUT Number2
         ELSE
            OUTPUT Number3
      ENDIF
ENDIF
```

Question 11.02

First part

```
INPUT BiggestSoFar
Counter ← 1
REPEAT
   INPUT NextNumber
   Counter ← Counter + 1
   IF NextNumber > BiggestSoFar
      THEN
         BiggestSoFar ← NextNumber
   ENDIF
UNTIL Counter = 100
OUTPUT BiggestSoFar
```

Second part

```
INPUT MaxNumbers
INPUT BiggestSoFar
Counter ← 1
REPEAT
   INPUT NextNumber
   Counter ← Counter + 1
   IF NextNumber > BiggestSoFar
      THEN
         BiggestSoFar ← NextNumber
   ENDIF
UNTIL Counter = MaxNumbers
OUTPUT BiggestSoFar
```

Task 11.04

```
RunningTotal ← 0
Count ← 0
REPEAT
   INPUT NextNumber
   RunningTotal ← RunningTotal + NextNumber
   Count ← Count + 1
UNTIL NextNumber = 0
OUTPUT RunningTotal
Average ← RunningTotal / (Count – 1)
OUTPUT Average
```

Task 11.05

Arrays with friends' names and ages

Task 11.06

```
MaxIndex ← 20
INPUT SearchValue
Found ← FALSE
Index ← 0
REPEAT
   Index ← Index + 1
```

```
   IF Name[Index]= SearchValue
      THEN
         Found ← TRUE
   ENDIF
UNTIL FOUND = TRUE OR Index >= MaxIndex
IF Found = TRUE
   THEN
      OUTPUT "Age:" Age[Index]
   ELSE
      OUTPUT "Name not found"
ENDIF
```

Task 11.07

```
n ← MaxIndex – 1
REPEAT
   NoMoreSwaps ← TRUE
   FOR j ← 1 TO n
      IF MyList[j] < MyList[j + 1]
         THEN
            Temp ← MyList[j]
            MyList[j]← MyList[j + 1]
            MyList[j + 1]← Temp
            NoMoreSwaps ← FALSE
      ENDIF
   ENDFOR
   n ← n – 1
UNTIL NoMoreSwaps = TRUE
```

Exam-style questions in Chapter 11

1

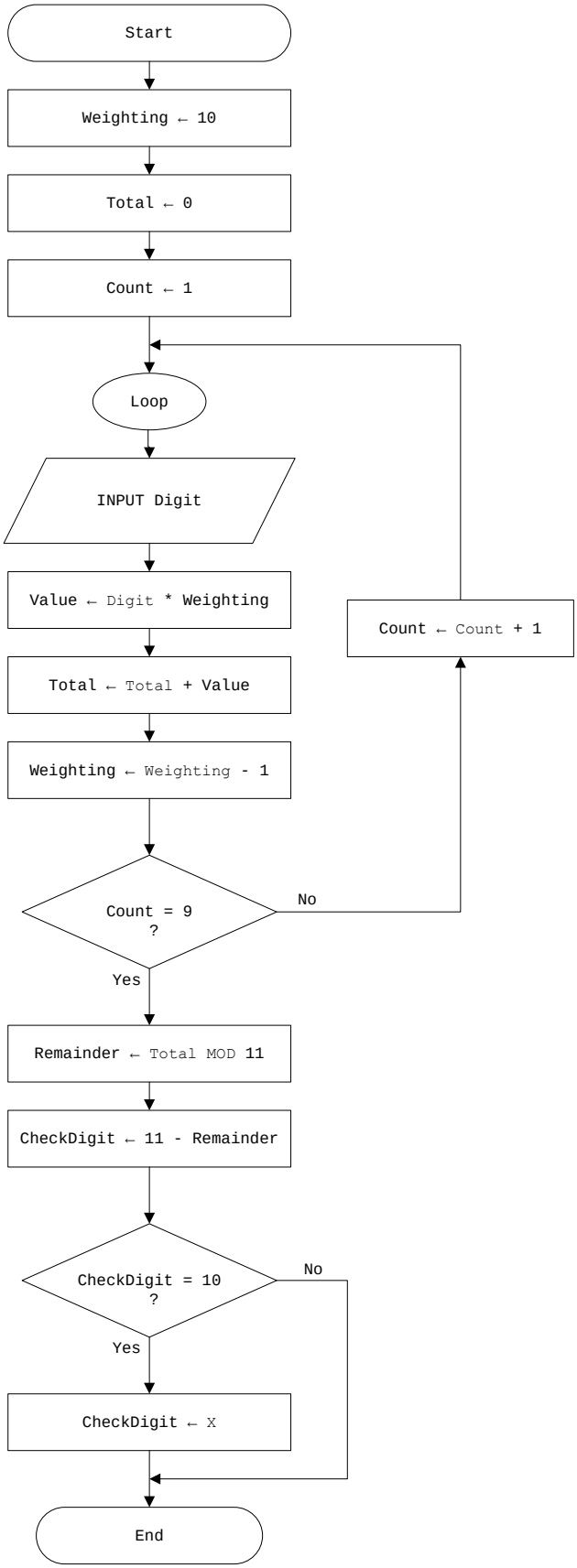

Figure 11.07

2

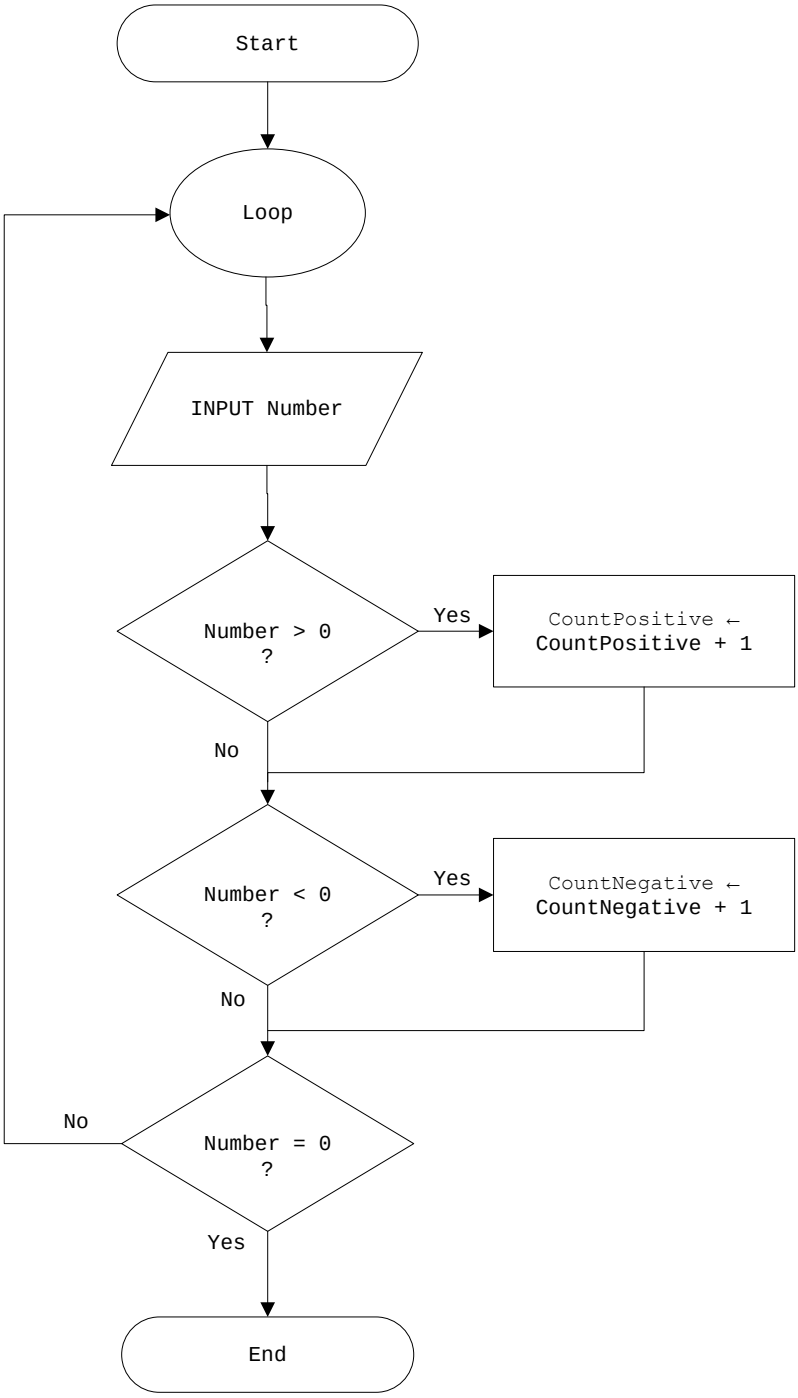

Figure 11.08

3

```
RogueValue← -1
Total ← 0
Count ← 0
INPUT Number
WHILE Number <>RogueValue
   Count ← Count + 1
   Total ← Total + Number
   INPUT Number
ENDWHILE
If Count > 0
   THEN
      Average ← Total / Count
      OUTPUT Average
ENDIF
```

4 a

Identifier	Explanation
UserList[1..20]	1D array to store user IDs
PasswordList[1..20]	1D array to store passwords
MaxIndex	Number of elements in each array
MyUserID	User ID entered to log in
MyPassword	Password entered to log in
UserIdFound	FALSE if user ID not found in UserList TRUE if user ID found in UserList
LoginOK	FALSE if user ID not found, or passwords don't match TRUE if password correctly entered for existing user ID
Index	Pointer to current list element

b

```
MaxIndex ← 20
INPUT MyUserID
INPUT MyPassword
UserIdFound ← FALSE
LoginOK ← FALSE
Index ← 0
REPEAT
   INDEX ← Index + 1
   IF UserList[Index] = MyUserID
      THEN
UserIdFound ← TRUE
   ENDIF
UNTIL  UserIdFound = TRUE
       OR Index = MaxIndex
IF UserIdFound = TRUE
   THEN
      IF PasswordList[Index] = MyPassword
         THEN
LoginOK ← TRUE
      ENDIF
ENDIF
IF LoginOK = TRUE
   THEN
```

```
      OUTPUT "Login successful"
   ELSE
      OUTPUT "User ID and/or password incorrect"
ENDIF
```

Question 13.01

```
4 * 3 – 3 ^ 2 = 3

(4 * 3 – 3) ^ 2 = 81

4 * (3 – 3) ^ 2 = 0

4 * (3 – 3 ^ 2) = -24
```

Task 13.01

Python

```python
# YourName : string data type
YourName = input("What is your name? ")
print("Have a nice day, ", YourName)
```

Figure 11.09

VB.NET

```
Module1.vb
(General)
Module Module1

    Sub Main()
        Dim YourName As String
        Console.Write("What is your name? ")
        YourName = Console.ReadLine()
        Console.WriteLine("Have a nice day, " & YourName)
        Console.ReadLine()
    End Sub

End Module
```

Figure 11.10

Pascal

```
Project2
program Project2;

{$APPTYPE CONSOLE}

uses
  SysUtils;
var
   YourName : string;
begin
  Write('What is your name? ');
  ReadLn(YourName);
  WriteLn('Have a nice day, ', YourName);
  ReadLn;
end.
```

	Figure 11.11

Task 13.02part 1

Identifier	Explanation	Data type
`Miles`	Distance as a whole number of miles	INTEGER
`Km`	The result from the conversion of `Miles`using the given formula:`Miles * 1.61`	REAL

Identifier	Explanation	Data type
`Number1`	The first number to be input	INTEGER/REAL
`Number2`	The second number to be input	INTEGER/REAL
`Number3`	The third number to be input	INTEGER/REAL

Identifier	Explanation	Data type
`BiggestSoFar`	Stores the biggest number input so far	INTEGER/REAL
`NextNumber`	The next number to be input	INTEGER/REAL

Identifier	Explanation	Data type
`BiggestSoFar`	Stores the biggest number input so far	INTEGER/REAL
`NextNumber`	The next number to be input	INTEGER/REAL
`Counter`	Stores how many numbers have been input so far	INTEGER

Identifier	Explanation	Data type
`SecretNumber`	The number to be guessed	INTEGER
`NumberOfGuesses`	The number of guesses the player has made	INTEGER

Guess	The number the player has input as a guess	INTEGER

Identifier	Explanation	Data type
RunningTotal	Stores the sum of the numbers input so far	INTEGER/REAL
Counter	How many numbers have been input	INTEGER
NextNumber	The next number input	INTEGER/REAL
Average	The average of the numbers input	REAL

Identifier	Explanation	Data type
NumberOfRows	Stores the number of rows of the grid	INTEGER
NumberOfColumns	Stores the number of columns of the grid	INTEGER
RowCounter	Counts the number of rows	INTEGER
ColumnCounter	Counts the number of columns	INTEGER
Symbol	Stores the chosen character symbol	CHAR

Identifier	Explanation	Data type
MyList	Data structure (1D array) to store seven numbers	ARRAY of INTEGER
MaxIndex	The number of elements in the array	INTEGER
SearchValue	The value to be searched for	INTEGER
Found	TRUEif the value has been found FALSEif the value has not been found	BOOLEAN
Index	Index of the array element currently being processed	INTEGER

Task 13.02 part 2

```
INPUT "Enter miles:" Miles
Km ← Miles * 1.61
```

```
OUTPUT "km:" Km
```

Python	`# Miles : int`
	`# Km : float`
	`Miles = int(input("Enter miles: "))`
	`Km = Miles * 1.61`
	`print("km: ", Km)`
		No variable declarations in Python. However, students should be encouraged to write comments at the beginning of the program to state the variables and the data types to be used. Note that Python only accepts string values as input. So a number must be converted with the built-in function int().

VB.NET	`ModuleModule1`
	`Dim Miles AsInteger`
	`Dim Km AsSingle`
	`Sub Main()`
	`Console.Write("Enter miles: ")`
	`  Miles = Console.ReadLine()`
	`  Km = Miles * 1.61`
	`Console.WriteLine("km: "& Km)`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`	

Pascal	`program Project1;`
	
	`{$APPTYPE CONSOLE}`
	
	`uses`
	`  SysUtils;`
	`  var Miles : Integer;`
	`  Km : real;`
	`begin`
	`  Write('Enter miles: ');`
	`  ReadLn(Miles);`
	`  Km := Miles * 1.61;`
	`  WriteLn('km: ', Km);`
	`  ReadLn;`
	`end.`	

Task 13.03 (Worked example 11.03)

```
INPUT BiggestSoFar
INPUT NextNumber
IF NextNumber>BiggestSoFar
   THEN
BiggestSoFar←NextNumber
ENDIF
INPUT NextNumber
IF NextNumber>BiggestSoFar
THEN
BiggestSoFar←NextNumber
ENDIF
```

OUTPUT BiggestSoFar

Python

```
BiggestSoFar = int(input("Enter first number: "))
NextNumber = int(input("Enter another number: "))
if NextNumber>BiggestSoFar:
BiggestSoFar = NextNumber
NextNumber = int(input("Enter another number: "))
if NextNumber>BiggestSoFar:
BiggestSoFar = NextNumber
print("biggest number is: ", BiggestSoFar)
```

VB.NET

```
ModuleModule1
DimBiggestSoFar, NextNumberAsInteger

Sub Main()
Console.Write("Enter first number: ")
BiggestSoFar = Console.ReadLine()
Console.Write("Enter another number: ")
NextNumber = Console.ReadLine()
IfNextNumber>BiggestSoFarThen
BiggestSoFar = NextNumber
EndIf
Console.Write("Enter another number: ")
NextNumber = Console.ReadLine()
IfNextNumber>BiggestSoFarThen
BiggestSoFar = NextNumber
EndIf
Console.WriteLine("biggest number is: "&BiggestSoFar)
Console.ReadLine()
EndSub

EndModule
```

Pascal

```
program Project2;

{$APPTYPE CONSOLE}

uses
SysUtils;

varBiggestSoFar, NextNumber : Integer;

begin
  Write('Enter first number: ');
ReadLn(BiggestSoFar);
  Write('Enter another number: ');
ReadLn(NextNumber);
  if NextNumber>BiggestSoFar
     then
BiggestSoFar := NextNumber;
  Write('Enter another number: ');
ReadLn(NextNumber);
  if NextNumber>BiggestSoFar
     then
BiggestSoFar := NextNumber;
WriteLn('biggest number is: ', BiggestSoFar);
ReadLn;
end.
```

Task 13.04 (Worked example 11.02)

```
INPUT Number1
INPUT Number2
INPUT Number3
IF Number1 > Number2
   THEN                        // Number1 is bigger
      IF Number1 > Number3
         THEN
            OUTPUT Number1
         ELSE
            OUTPUT Number3
      ENDIF
   ELSE                        // Number2 is bigger
      IF Number2 > Number3
         THEN
            OUTPUT Number2
         ELSE
            OUTPUT Number3
      ENDIF
ENDIF
```

Python

```
Number1 = int(input("Enter first number: "))
Number2 = int(input("Enter second number: "))
Number3 = int(input("Enter third number: "))
if Number1 > Number2:
   if Number1 > Number3:
      print(Number1)
   else:
      print(Number3)
elif Number2 > Number3:
         print(Number2)
else:
         print(Number3)
```

VB.NET

```
ModuleModule1
Dim Number1, Number2, Number3 AsInteger

Sub Main()
Console.Write("Enter first number: ")
        Number1 = Console.ReadLine()
Console.Write("Enter second number: ")
        Number2 = Console.ReadLine()
Console.Write("Enter third number: ")
        Number3 = Console.ReadLine()
If Number1 > Number2 Then
If Number1 > Number3 Then
Console.WriteLine(Number1)
Else
Console.WriteLine(Number3)
EndIf
ElseIf Number2 > Number3 Then
Console.WriteLine(Number2)
Else
Console.WriteLine(Number3)
```

	EndIf Console.ReadLine() EndSub EndModule

Pascal	program Project1;
	
	{$APPTYPE CONSOLE}
	
	uses
	SysUtils;
	var Number1, Number2, Number3 : Integer;
	begin
	Write('Enter first number: ');
	ReadLn(Number1);
	Write('Enter second number: ');
	ReadLn(Number2);
	Write('Enter third number: ');
	ReadLn(Number3);
	if Number1 > Number2
	then
	if Number1 > Number3
	then
	write(Number1)
	else
	write(Number3)
	
	else
	if Number2 > Number3
	then
	write(Number2)
	else
	write(Number3);
	ReadLn;
	end.

Task 13.05

```
INPUT MonthNumber
INPUT Year
Days ← 0
CASE OF MonthNumber
   CASE 1,3,5,7,8,10,12: Days ← 31
   CASE 4,6,9,11: Days ← 30
   CASE 2: Days ← 28
           If Year MOD 400 = 0
THEN  // it is a leap year
                 Days ← 29
           ENDIF
           IF (Year MOD 4 = 0) AND (Year MOD 100 > 0)
THEN  // it is a leap year
```

```
                Days ← 29
            ENDIF
   OTHERWISE: OUTPUT "Invalid Month number"
ENDCASE
OUTPUT Days
```

Python

```
MonthNumber = int(input("Enter month number: "))
Year = int(input("Enter year: "))
Days = 0
if MonthNumber in [1,3,5,7,8,10,12]:
   Days = 31
elifMonthNumber in [4,6,9,11]:
   Days = 30
elifMonthNumber == 2:
   Days = 28
   if Year % 400 == 0:
      Days = 29
   if Year % 4 == 0 and Year % 100 > 0:
      Days = 29
else:
   print("Invalid month number")
print(Days)
```

VB.NET

```
ModuleModule1
DimMonthNumber, Year, Days AsInteger

Sub Main()
Console.Write("Enter month number: ")
MonthNumber = Console.ReadLine()
Console.Write("Enter year: ")
        Year = Console.ReadLine()
        Days = 0
SelectCaseMonthNumber
Case 1, 3, 5, 7, 8, 10, 12
                Days = 31
Case 4, 6, 9, 11
                Days = 30
Case 2
                Days = 28
If Year Mod 400 = 0 Then
                    Days = 29
EndIf
If Year Mod 4 = 0 And Year Mod 100 > 0 Then
                    Days = 29
EndIf
Case Else
Console.WriteLine("invalid month number")
EndSelect
Console.WriteLine(Days)
Console.ReadLine()
EndSub
```

Pascal

```
varMonthNumber, Year, Days : Integer;

begin
  Write('Enter month number: ');
```

```
ReadLn(MonthNumber);
  Write('Enter Year YYYY: ');
ReadLn(Year);
  Days := 0;
  case MonthNumber of
     1,3,5,7,8,10,12: Days := 31;
     4,6,9,11: Days := 30;
     2: begin
           Days := 28;
           if Year mod 400 = 0
              then
                 Days := 29;
           if (Year mod 4 = 0) and (Year mod 100 > 0)
              then
                 Days := 29;
        end;
     else
        Write('Invalid month number');
  end;
  Write(Days);
ReadLn;
end.
```

Task 13.06 part 1

```
INPUT BiggestSoFar
FOR Counter ← 2 TO 10
   INPUT NextNumber
   IF NextNumber>BiggestSoFar
      THEN
BiggestSoFar←NextNumber
   ENDIF
ENDFOR
OUTPUT BiggestSoFar
```

Python

```
BiggestSoFar = int(input("Enter first number: "))
for Counter in range(2,11):
NextNumber = int(input("Enter another number: "))
   if NextNumber>BiggestSoFar:
BiggestSoFar = NextNumber
print(BiggestSoFar)
```

VB.NET

```
ModuleModule1
DimBiggestSoFar, Counter, NextNumberAsInteger

Sub Main()
Console.Write("Enter first number: ")
BiggestSoFar = Console.ReadLine()
For Counter = 2 To 10
Console.Write("Enter another number: ")
NextNumber = Console.ReadLine()
IfNextNumber>BiggestSoFarThen
BiggestSoFar = NextNumber
EndIf
Next
```

	`Console.WriteLine(BiggestSoFar)`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`varBiggestSoFar, NextNumber, Counter : Integer;`
	
	`begin`
	`  Write('Enter first number: ');`
	`ReadLn(BiggestSoFar);`
	`  for Counter := 2 to 10 do`
	`  begin`
	`  Write('Enter another number: ');`
	`ReadLn(NextNumber);`
	`  if NextNumber>BiggestSoFar`
	`  then`
	`BiggestSoFar := NextNumber;`
	`  end;`
	`  Write(BiggestSoFar);`
	`ReadLn;`
	`end.`

Task 13.06 part 2

```
RunningTotal← 0
FOR Counter ←1 TO 10
   INPUT NextNumber
RunningTotal←RunningTotal + NextNumber
ENDFOR
OUTPUT RunningTotal
Average ←RunningTotal / 10
OUTPUT Average
```

Python	`RunningTotal = 0`
	`for Counter in range(1,11):`
	`NextNumber = int(input("Enter a number: "))`
	`RunningTotal = RunningTotal + NextNumber`
	`print(RunningTotal)`
	`Average = RunningTotal / 10`
	`print(Average)`

VB.NET	`ModuleModule1`
	`DimRunningTotal, Counter, NextNumberAsInteger`
	`Dim Average AsSingle`
	
	`Sub Main()`
	`RunningTotal = 0`
	`For Counter = 1 To 10`
	`Console.Write("Enter a number: ")`
	`NextNumber = Console.ReadLine()`
	`RunningTotal = RunningTotal + NextNumber`

	`Next`
	`Console.WriteLine(RunningTotal)`
	`  Average = RunningTotal / 10`
	`Console.WriteLine(Average)`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`varRunningTotal, NextNumber, Counter : Integer;`
	`  Average : Real;`
	
	`begin`
	`RunningTotal := 0;`
	`  for Counter := 1 to 10 do`
	`  begin`
	`  Write('Enter a number: ');`
	`ReadLn(NextNumber);`
	`RunningTotal := RunningTotal + NextNumber;`
	`  end;`
	`  Write(RunningTotal);`
	`  Average := RunningTotal / 10;`
	`  Write(Average);`
	`ReadLn;`
	`end.`

Task 13.06 part 3

```
INPUT NumberOfRows
INPUT NumberOfColumns
INPUT Symbol
FOR RowCounter←1 TO NumberOfRows
   FOR ColumnCounter← 1 TO NumberOfColumns
      OUTPUT Symbol           // without moving to next line
   ENDFOR
   OUTPUT Newline             // move to the next line
ENDFOR
```

Python	`NumberOfRows = int(input("Enter number of rows: "))`
	`NumberOfColumns = int(input("Enter number of columns: "))`
	`Symbol = input("Enter symbol: ")`
	`for RowCounter in range(0,NumberOfRows):`
	`  for ColumnCounter in range(0, NumberOfColumns):`
	`  print(Symbol, end='')`
	`  print()`

VB.NET	`ModuleModule1`
	`DimNumberOfRows, NumberOfColumns, RowCounter, ColumnCounterAsInteger`
	`Dim Symbol AsChar`
	
	`Sub Main()`
	`Console.Write("Enter number of rows: ")`
	`NumberOfRows = Console.ReadLine()`

	`Console.Write("Enter number of columns: ")`
	`NumberOfColumns = Console.ReadLine()`
	`Console.Write("Enter symbol: ")`
	`  Symbol = Console.ReadLine()`
	`ForRowCounter = 1 ToNumberOfRows`
	`ForColumnCounter = 1 ToNumberOfColumns`
	`Console.Write(Symbol)`
	`Next`
	`Console.WriteLine()`
	`Next`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`varNumberOfRows, NumberOfColumns, RowCounter,`
	`ColumnCounter : Integer;`
	`  Symbol : Char;`
	
	`begin`
	`  Write('Enter number of rows: ');`
	`ReadLn(NumberOfRows);`
	`  Write('Enter number of columns: ');`
	`ReadLn(NumberOfColumns);`
	`  Write('Enter symbol: ');`
	`ReadLn(Symbol);`
	`  for RowCounter := 1 to NumberOfRows do`
	`  begin`
	`  for ColumnCounter := 1 to NumberOfColumns do`
	`  Write(Symbol);`
	`Writeln;`
	`  end;`
	`ReadLn;`
	`end.`

Task 13.07 part 1

```
INPUT BiggestSoFar
Counter ← 1
REPEAT
   INPUT NextNumber
   Counter ←Counter + 1
   IF NextNumber>BiggestSoFar
      THEN
BiggestSoFar←NextNumber
   ENDIF
UNTIL Counter = 10
OUTPUT BiggestSoFar
```

Python	`BiggestSoFar = int(input("Enter first number: "))`
	`Counter = 1`
	`while Counter < 10:`
	`NextNumber = int(input("Enter another number: "))`

	`Counter = Counter + 1`
	`if NextNumber>BiggestSoFar:`
	`BiggestSoFar = NextNumber`
	`print(BiggestSoFar)`

VB.NET	`ModuleModule1`
	`DimBiggestSoFar, Counter, NextNumberAsInteger`
	
	`Sub Main()`
	`Console.Write("Enter first number: ")`
	`BiggestSoFar = Console.ReadLine()`
	`Counter = 1`
	`Do`
	`Console.Write("Enter another number: ")`
	`NextNumber = Console.ReadLine()`
	`Counter = Counter + 1`
	`IfNextNumber>BiggestSoFarThen`
	`BiggestSoFar = NextNumber`
	`EndIf`
	`LoopUntil Counter = 10`
	`Console.WriteLine(BiggestSoFar)`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`varBiggestSoFar, NextNumber, Counter : Integer;`
	
	`begin`
	`Write('Enter first number: ');`
	`ReadLn(BiggestSoFar);`
	`Counter := 1;`
	`repeat`
	`Write('Enter another number: ');`
	`ReadLn(NextNumber);`
	`Counter := Counter + 1;`
	`if NextNumber>BiggestSoFar`
	`then`
	`BiggestSoFar := NextNumber;`
	`until Counter = 10;`
	`Write(BiggestSoFar);`
	`ReadLn;`
	`end.`
	
	`end.`

Task 13.07 part 2

```
INPUT BiggestSoFar
REPEAT
   INPUT NextNumber
   IF NextNumber>BiggestSoFar
      THEN
BiggestSoFar←NextNumber
```

```
   ENDIF
UNTIL NextNumber = 0
OUTPUT BiggestSoFar
```

Python	`NextNumber = int(input("Enter first number: "))`
	`BiggestSoFar = NextNumber`
	`while NextNumber != 0:`
	`NextNumber = int(input("Enter another number: "))`
	`  if NextNumber>BiggestSoFar:`
	`BiggestSoFar = NextNumber`
	`print(BiggestSoFar)`

VB.NET	`ModuleModule1`
	`DimBiggestSoFar, NextNumberAsInteger`
	
	`Sub Main()`
	`Console.Write("Enter first number: ")`
	`BiggestSoFar = Console.ReadLine()`
	`Do`
	`Console.Write("Enter another number: ")`
	`NextNumber = Console.ReadLine()`
	`IfNextNumber>BiggestSoFarThen`
	`BiggestSoFar = NextNumber`
	`EndIf`
	`LoopUntilNextNumber = 0`
	`Console.WriteLine(BiggestSoFar)`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`varBiggestSoFar, NextNumber : Integer;`
	
	`begin`
	`  Write('Enter first number: ');`
	`ReadLn(BiggestSoFar);`
	`  repeat`
	`  Write('Enter another number: ');`
	`ReadLn(NextNumber);`
	`  if NextNumber>BiggestSoFar`
	`  then`
	`BiggestSoFar := NextNumber;`
	`  until NextNumber = 0;`
	`  Write(BiggestSoFar);`
	`ReadLn;`
	`end.`

Task 13.08

```
INPUT NextNumber
BiggestSoFar←NextNumber
WHILE NextNumber<> 0   // sequence terminator not encountered
   INPUT NextNumber
   IF NextNumber>BiggestSoFar
```

```
      THEN
BiggestSoFar←NextNumber
   ENDIF
ENDWHILE
OUTPUT BiggestSoFar
```

Python

```
NextNumber = int(input("Enter first number: "))
BiggestSoFar = NextNumber
while NextNumber != 0:
NextNumber = int(input("Enter another number: "))
   if NextNumber>BiggestSoFar:
BiggestSoFar = NextNumber
print(BiggestSoFar)
```

VB.NET

```
ModuleModule1
DimBiggestSoFar, NextNumberAsInteger

Sub Main()
Console.Write("Enter first number: ")
NextNumber = Console.ReadLine()
BiggestSoFar = NextNumber
DoWhileNextNumber<> 0
Console.Write("Enter another number: ")
NextNumber = Console.ReadLine()
IfNextNumber>BiggestSoFarThen
BiggestSoFar = NextNumber
EndIf
Loop
Console.WriteLine(BiggestSoFar)
Console.ReadLine()
EndSub

EndModule
```

Pascal

```
varBiggestSoFar, NextNumber : Integer;

begin
  Write('Enter first number: ');
ReadLn(NextNumber);
BiggestSoFar := NextNumber;
  while NextNumber<> 0 do
     begin
        Write('Enter another number: ');
ReadLn(NextNumber);
        if NextNumber>BiggestSoFar
           then
BiggestSoFar := NextNumber;
     end;
  Write(BiggestSoFar);
ReadLn;
end.
```

Task 13.09 part 1

```
FOR Index ← 1 TO 7
   INPUT MyList[Index]
ENDFOR
```

Python	`MyList = []`
	`for Index in range(8):`
	`MyList.append( int(input("Enter a number: ")))`
	`for Index in range(1,8):`
	`  print(MyList[Index])`

VB.NET	`ModuleModule1`
	`DimMyList(7) AsInteger`
	`Dim Index AsInteger`
	
	`Sub Main()`
	`For Index = 1 To 7`
	`Console.Write("Enter a number: ")`
	`MyList(Index) = Console.ReadLine()`
	`Next`
	`For Index = 1 To 7`
	`Console.Write(MyList(Index) &" ")`
	`Next`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`varMyList : Array[1..7] of integer;`
	`  Index : Integer;`
	
	`begin`
	`  for Index := 1 to 7 do`
	`  begin`
	`  Write('Enter a number: ');`
	`ReadLn(MyList[Index]);`
	`  end;`
	
	`  for Index := 1 to 7 do`
	`  begin`
	`  Write(MyList[Index], '  ');`
	`  end;`
	`ReadLn;`
	`end.`

Task 13.09 part 2

```
MaxIndex← 7
INPUT SearchValue
Found ← FALSE
Index ←0
REPEAT
   Index ←Index + 1
```

```
   IF MyList[Index] = SearchValue
      THEN
         Found ← TRUE
   ENDIF
UNTIL FOUND = TRUE OR Index >=MaxIndex
IF Found = TRUE
   THEN
      OUTPUT "Value found at location:" Index
   ELSE
      OUTPUT "Value not found"
ENDIF
```

Python

```
MyList = []
for Index in range(7):
MyList.append( int(input("Enter a number: ")))
MaxIndex = 6
SearchValue = int(input("Enter search value: "))
Found = False
Index = 0
while not Found and Index <MaxIndex:
   Index = Index + 1
   if MyList[Index] == SearchValue:
      Found = True
if Found:
   print("Value found at location: ", Index+1)
else:
   print("Value not found")
```

VB.NET

```
ModuleModule1
DimMyList(7) AsInteger
Dim Index, MaxIndex, SearchValueAsInteger
Dim Found AsBoolean

Sub Main()
For Index = 1 To 7
Console.Write("Enter a number: ")
MyList(Index) = Console.ReadLine()
Next
For Index = 1 To 7
Console.Write(MyList(Index) &" ")
Next
MaxIndex = 7
Console.Write("Enter search value: ")
SearchValue = Console.ReadLine()
        Found = False
        Index = 0
Do
            Index = Index + 1
IfMyList(Index) = SearchValueThen
                Found = True
EndIf
LoopUntil Found Or Index >= MaxIndex
If Found Then
Console.WriteLine("Value found at location "& Index)
Else
Console.WriteLine("Value not found")
```

	EndIf Console.ReadLine() EndSub EndModule

Pascal	varMyList : Array[1..7] of integer;
	Index, MaxIndex, SearchValue : Integer;
	Found : Boolean;
	
	begin
	MaxIndex := 7;
	write('Enter search value: ');
	ReadLn(SearchValue);
	Found := False;
	Index := 0;
	repeat
	Index := Index + 1;
	if MyList[Index] = SearchValue
	then
	Found := True;
	until Found OR (Index >=MaxIndex);
	if Found
	then
	write('value found at location ', Index)
	else
	write('Value not found');
	ReadLn;

Task 13.09 part 3

```
n←MaxIndex - 1
REPEAT
NoMoreSwaps← TRUE
   FOR j ←1 TO n
      IF MyList[j] >MyList[j + 1]
         THEN
            Temp ←MyList[j]
MyList[j] ←MyList[j + 1]
MyList[j + 1] ←Temp
NoMoreSwaps←FALSE
      ENDIF
   ENDFOR
n←n - 1
UNTIL NoMoreSwaps = TRUE
```

Python	MyList = [] for Index in range(7): MyList.append(int(input("Enter a number: ")))

	`MaxIndex = 7`
	`n = MaxIndex - 1`
	`NoMoreSwaps = False`
	`while NoMoreSwaps == False:`
	`NoMoreSwaps = True`
	`  for j in range(n):`
	`  if MyList[j] >MyList[j + 1]:`
	`  Temp = MyList[j]`
	`MyList[j] = MyList[j + 1]`
	`MyList[j + 1] = Temp`
	`NoMoreSwaps = False`
	`  n = n - 1`
	
	`for Index in range(7):`
	`  print(MyList[Index])`

VB.NET	`ModuleModule1`
	`DimMyList(7) AsInteger`
	`Dim Index, MaxIndex, n, j, Temp AsInteger`
	`DimNoMoreSwapsAsBoolean`
	
	`Sub Main()`
	`For Index = 1 To 7`
	`Console.Write("Enter a number: ")`
	`MyList(Index) = Console.ReadLine()`
	`Next`
	
	`MaxIndex = 7`
	`  n = MaxIndex - 1`
	`Do`
	`NoMoreSwaps = True`
	`For j = 1 To n`
	`IfMyList(j) >MyList(j + 1) Then`
	`  Temp = MyList(j)`
	`MyList(j) = MyList(j + 1)`
	`MyList(j + 1) = Temp`
	`NoMoreSwaps = False`
	`EndIf`
	`Next`
	`  n = n - 1`
	`LoopUntilNoMoreSwaps`
	
	`For Index = 1 To 7`
	`Console.Write(MyList(Index) &" ")`
	`Next`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`varMyList : Array[1..7] of integer;`
	`  Index, MaxIndex, n, j, Temp : Integer;`
	`NoMoreSwaps : Boolean;`
	
	`begin`
	`  for Index := 1 to 7 do`
	`  begin`
	`  Write('Enter a number: ');`
	`ReadLn(MyList[Index]);`

```
     end;
MaxIndex := 7;
  n := MaxIndex - 1;
  repeat
NoMoreSwaps := True;
     for j := 1 to n do
        if MyList[j] >MyList[j + 1]
           then
              begin
                 Temp := MyList[j];
MyList[j] := MyList[j + 1];
MyList[j + 1] := Temp;
NoMoreSwaps := False;
              end;
     n := n - 1;
  until NoMoreSwaps;
  for Index := 1 to 7 do
        write(MyList[Index], '  ');
ReadLn;
end.

varMyList : Array[1..7] of integer;
    Index, MaxIndex, n, j, Temp : Integer;
NoMoreSwaps : Boolean;
```

Task 13.10

```
FOR Row ←1 TO MaxRows
   FOR Column ←1 TO MaxColumns
      OUTPUT ThisTable[Row, Column] // stay on same line
   ENDFOR
   OUTPUT Newline                   // move to next line for
next row
ENDFOR
```

Python

```
ThisTable = [[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0]]
MaxRows = 6
MaxColumns = 7
for Row in range(MaxRows):
   for Column in range(MaxColumns):
      print(ThisTable[Row][Column], end = ' ')
   print()
```

VB.NET

```
ModuleModule1
DimThisTable(6, 7) AsInteger
Dim Row, Column AsInteger
```

	``` ConstMaxRows = 6 ConstMaxColumns = 7 Sub Main() For Row = 1 ToMaxRows For Column = 1 ToMaxColumns ThisTable(Row, Column) = 0 Next Next For Row = 1 ToMaxRows For Column = 1 ToMaxColumns Console.Write(ThisTable(Row, Column)) Next Console.WriteLine() Next Console.ReadLine() EndSub EndModule ```

Pascal	```
	const
	MaxRows = 6;
	MaxColumns = 7;
	
	varThisTable : Array[1..6, 1..7] of integer;
	Row, Column : Integer;
	
	begin
	for Row := 1 to MaxRows do
	for Column := 1 to MaxColumns do
	ThisTable[Row, Column] := 0;
	
	for Row := 1 to MaxRows do
	begin
	for Column := 1 to MaxColumns do
	Write(ThisTable[Row, Column]);
	WriteLn;
	end;
	
	ReadLn;
	end.
	``

Task 13.11 part 1

Python	```
	import random
	for Count in range(20):
	RandomNumber = random.randint(1,10)
	print(RandomNumber)
	``

VB.NET	``` ModuleModule1 Dim Count, RandomNumberAsInteger Dim Number AsNewRandom ```

	`Sub Main()`
	`For Count = 1 To 20`
	`RandomNumber = Number.Next(1, 11)`
	`Console.WriteLine(RandomNumber)`
	`Next`
	
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`uses`
	`SysUtils, Math;`
	
	`var Count, RandomNumber : Integer;`
	
	`begin`
	`  randomize;`
	`  for Count := 1 To 20 do`
	`  begin`
	`RandomNumber := RandomRange(1, 11);`
	`writeln(RandomNumber);`
	`  end;`
	
	`ReadLn;`
	`end.`

Task 13.11 part 2

```
SecretNumber←Random
INPUT Guess
NumberOfGuesses←1
WHILE Guess <>SecretNumber AND NumberOfGuesses< 10
   IF Guess >SecretNumber
      THEN
the player is given the message to input a smaller number
   ENDIF
   IF Guess <SecretNumber
      THEN
the player is given the message to input a larger number
   ENDIF
   INPUT Guess
NumberOfGuesses←NumberOfGuesses + 1
ENDWHILE
```

Python	`import random`
	`SecretNumber = random.randint(1,100)`
	`Guess = int(input("What is your guess? "))`
	`NumberOfGuesses = 1`

	`while Guess != SecretNumber and NumberOfGuesses< 10:`
	`  if Guess >SecretNumber:`
	`  print("Input a smaller number: ", end='')`
	`  if Guess <SecretNumber:`
	`  print("Input a larger number: ", end='')`
	`  Guess = int(input())`
	`NumberOfGuesses = NumberOfGuesses + 1`
	`if Guess == SecretNumber:`
	`print("Well done. It took you ",NumberOfGuesses, "`
	`guesses.")`
	`else:`
	`print("You did not guess the number. It was ",`
	`SecretNumber)`

VB.NET	`ModuleModule1`
	`Dim Guess, SecretNumber, NumberOfGuessesAsInteger`
	`Dim Number AsNewRandom`
	
	`Sub Main()`
	
	`SecretNumber = Number.Next(1, 100)`
	`Console.Write("What is your guess? ")`
	`  Guess = Console.ReadLine()`
	`NumberOfGuesses = 1`
	`While Guess <>SecretNumberAndNumberOfGuesses< 10`
	`If Guess >SecretNumberThen`
	`Console.Write("Input a smaller number: ")`
	`EndIf`
	`If Guess <SecretNumberThen`
	`Console.Write("Input a larger number: ")`
	`EndIf`
	`  Guess = Console.ReadLine()`
	`NumberOfGuesses = NumberOfGuesses + 1`
	`EndWhile`
	`If Guess = SecretNumberThen`
	`Console.WriteLine("Well done. It took you {0} guesses.",`
	`NumberOfGuesses)`
	`Else`
	`Console.WriteLine("You did not guess the number. It was {0}.",`
	`SecretNumber)`
	`EndIf`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`uses`
	`SysUtils, Math;`
	
	`var Guess, NumberOfGuesses, SecretNumber : Integer;`
	
	`begin`
	`  randomize;`
	`SecretNumber := randomrange(1,100);`
	`Write('What is your guess? ');`
	`ReadLn(Guess);`
	`NumberOfGuesses := 1;`
	`  while (Guess <>SecretNumber) and (NumberOfGuesses<`

	10) do
	begin
	if Guess >SecretNumber
	then
	Write('Input a smaller number: ')
	else
	Write('Input a larger number: ');
	ReadLn(Guess);
	NumberOfGuesses := NumberOfGuesses + 1;
	end;
	if Guess = SecretNumber
	then
	WriteLn('Well done. It took you ', NumberOfGuesses, ' guesses.')
	else
	WriteLn('You did not guess the number. It was ', SecretNumber);
	ReadLn;
	end.

Task 13.12

Python	from datetime import * Today = date.today() print('Today is ', Today) Tomorrow = Today + timedelta(1) print('Tomorrow is ', Tomorrow) Yesterday = Today + timedelta(-1) print('Yesterday was ', Yesterday)
VB.NET	ModuleModule1 Dim Today, Tomorrow, Yesterday AsDate Sub Main() Today = Now() Console.WriteLine("Today is {0:dd/MM/yy}", Today) Tomorrow = Today.AddDays(1) Console.WriteLine("Tomorrow is {0:dd/MM/yy}", Tomorrow) Yesterday = Today.AddDays(-1) Console.WriteLine("Yesterday was {0:dd/MM/yy}", Yesterday) Console.ReadLine() EndSub EndModule
Pascal	var Today, Tomorrow, Yesterday : TDateTime; DateString : String; begin Today := Date(); DateString := DateToStr(Today); WriteLn('Today is ', DateString);

	`  Tomorrow := Today + 1;`
	`DateString := DateToStr(Tomorrow);`
	`WriteLn('Tomorrow is ', DateString);`
	`  Yesterday := Today - 1;`
	`DateString := DateToStr(Yesterday);`
	`WriteLn('Yesterday was ', DateString);`
	`ReadLn;`
	`end.`

Exam-style questionsin Chapter 13

1

```
OUTPUT "Ounces       Grams"
FOR Ounces ← 1 TO 30
   Grams ←Rounded(Ounces * 28.35)// whole number of grams
only
   OUTPUT Ounces, Grams
ENDFOR
```

Python	`# Ounces : int`
	`# Grams : int`
	`print('Ounces       Grams')`
	`for Ounces in range(1, 31):`
	`  Grams = round(Ounces * 28.35) # whole number of grams only`
	`  print("{0:>4}{1:>13}".format(Ounces, Grams))`

VB.NET	`ModuleModule1`
	
	`Sub Main()`
	`Dim Ounces, Grams AsInteger`
	`Console.WriteLine("Ounces       Grams")`
	`For Ounces = 1 To 30`
	`  Grams = Math.Round(Ounces * 28.35)`
	`Console.WriteLine("{0,4}{1,13}", Ounces, Grams)`
	`Next`
	`Console.ReadLine()`
	`EndSub`
	
	`EndModule`

Pascal	`program Project1;`
	
	`{$APPTYPE CONSOLE}`
	
	`uses`
	`SysUtils;`
	`var Ounces, Grams : integer;`
	`begin`
	`WriteLn('Ounces       Grams');`
	`  for Ounces := 1 to 30 do`
	`  begin`

```
            Grams := round(Ounces * 28.35);
WriteLn(Ounces : 4, Grams : 13);
      end;
ReadLn;
end.
```

2

Python

```
# UserID : str
# valid : bool
UserID = input('Enter your user ID: ')
if len(UserID) != 5:
   valid = False
else:
   valid = True
   for Char in range(3):
      if UserID[Char] < 'A' or UserID[Char] > 'Z':
         valid = False
   for Char in range(3,5):
      if UserID[Char] < '0' or UserID[Char] > '9':
         valid = False

if valid:
   print("valid")
else:
   print("not valid")
```

VB.NET

```
ModuleModule1

Sub Main()
DimUserIDAsString
Dim Valid AsBoolean
DimiAsInteger
Console.Write("Enter your user ID: ")
UserID = Console.ReadLine

If Len(UserID) <> 5 Then
            Valid = False
Else
            valid = True
Fori = 0 To 2
IfUserID(i) <"A"OrUserID(i) >"Z"Then
                    Valid = False
EndIf
Next
Fori = 3 To 4
IfUserID(i) <"0"OrUserID(i) >"9"Then
                    Valid = False
EndIf
Next
EndIf
If Valid Then
Console.WriteLine("valid")
Else
Console.WriteLine("not valid")
EndIf
Console.ReadLine()
EndSub
```

	EndModule

Pascal	program Project1;
	
	{$APPTYPE CONSOLE}
	
	uses
	SysUtils;
	varUserID : string;
	  Valid : Boolean;
	i : integer;
	begin
	  Write('Enter your user ID: ');
	ReadLn(UserID);
	  if Length(UserID) <> 5
	  then Valid := False
	  else
	  begin
	  Valid := True;
	  for i := 1 to 3 do
	  if (UserID[i] < 'A') or (UserID[i] > 'Z')
	  then Valid := False;
	  for i := 4 to 5 do
	  if (UserID[i] < '0') or (UserID[i] > '9')
	  then Valid := False;
	  end;
	  if valid
	  then WriteLn('valid')
	  else WriteLn('not valid');
	ReadLn;
	end.

3

```
Initialise Tally array
REPEAT
INPUT Choice // 1 for Reading, 2 for computer games,
                // 3 for Sport, 4 for Programming, 5 for TV
// 0 to end input
Increment Tally[Choice]
UNTIL Choice = 0
FOR Index = 1 TO 5
   OUTPUT Tally[Index]
ENDFOR
```

Python	# Tally : int array [1:5] # Hobbytitle : str array [1:5] # Choice : int # Index : int # FileHandle : text file channel

```
# TextLine : str
# DigitString : str

# (a) declare and initialise Tally
Tally = [0,0,0,0,0,0,0]

# (c) store hobby titles
HobbyTitle = ['','Reading books','Playing computer
games', 'Sport', 'Programming', 'Watching TV']

# (e) to read existing data from file
FileHandle = open("Tally.TXT", "r")
for Index in range(6):
TextLine = FileHandle.readline()   # read next line
DigitString = TextLine.strip('\n') # strip newline
character
   Tally[Index] = int(DigitString)    # convert str to
int
FileHandle.close()

# (b) main part of program
Choice = int(input('Enter your favourite hobby: '))
while Choice != 0:
   Tally[Choice] += 1
   Choice = int(input('Enter your favourite hobby: '))
for Index in range(1,6):
   print("{0:<24}{1:>5}".format(HobbyTitle[Index],
Tally[Index]))

# (d) save tally data to file
FileHandle = open("Tally.TXT", "w")
for Index in range(6):
FileHandle.write(str(Tally[Index]) + "\n")
FileHandle.close()
```

VB.NET

```
ModuleModule1

Sub Main()
Dim Tally(5) AsInteger
DimHobbytitle(5) AsString
Dim Choice, Index AsInteger
DimFileHandleOutAsIO.StreamWriter
DimFileHandleInAsIO.StreamReader

' (a) declare and initialise Tally
For Index = 0 To 5
            Tally(Index) = 0
Next

' (c) store hobby titles
Hobbytitle(1) = "Reading books"
Hobbytitle(2) = "Playing computer games"
Hobbytitle(3) = "Sport"
Hobbytitle(4) = "Programming"
Hobbytitle(5) = "Watching TV"
```

```
' (e) to read existing data from file
FileHandleIn = NewIO.StreamReader("Tally.TXT")
For Index = 0 To 5
            Tally(Index) = FileHandleIn.ReadLine
Next
FileHandleIn.Close()

' (b) main part of program
Do
Console.Write("Enter your favourite hobby: ")
            Choice = Console.ReadLine
            Tally(Choice) += 1
LoopUntil Choice = 0
For Index = 1 To 5
Console.WriteLine("{0,24}{1,5}", Hobbytitle(Index), Tally(Index))
Next

' (d) save tally data to file
FileHandleOut = NewIO.StreamWriter("Tally.TXT")
For Index = 0 To 5
FileHandleOut.WriteLine(Tally(Index))
Next
FileHandleOut.Close()

Console.ReadLine()
EndSub

EndModule
```

Pascal

```
program ExamQ13_3;

{$APPTYPE CONSOLE}

uses
SysUtils;

var Tally : array[0..5] of integer;
Hobbytitle : array[1..5] of string;
      Choice, Index : integer;
FileHandle : TextFile;
begin
 // (a) declare and initialise Tally
 for Index := 0 to 5 do
   Tally[Index] := 0;

 // (c) store hobby titles
HobbyTitle[1] := 'Reading books';
HobbyTitle[2] := 'Playing computer games';
HobbyTitle[3] := 'Sport';
HobbyTitle[4] := 'Programming';
HobbyTitle[5] := 'Watching TV';

  // (e) to read existing data from file
AssignFile(FileHandle, 'Tally.TXT');
  Reset(FileHandle);
  for Index := 0 to 5 do
ReadLn(FileHandle, Tally[Index]);
CloseFile(FileHandle);
```

```
  // (b) main part of program
  repeat
     Write('Enter your favourite hobby: ');
ReadLn(Choice);
     Tally[Choice] := Tally[Choice] + 1;
  until Choice = 0;
  for Index := 1 to 5 do
WriteLn(Hobbytitle[Index]:24, Tally[Index]:5);

  // (d) save tally data to file
AssignFile(FileHandle, 'Tally.TXT');
  Rewrite(FileHandle);
  for Index := 0 to 5 do
WriteLn(FileHandle,Tally[Index]);
CloseFile(FileHandle);
ReadLn
end.
```