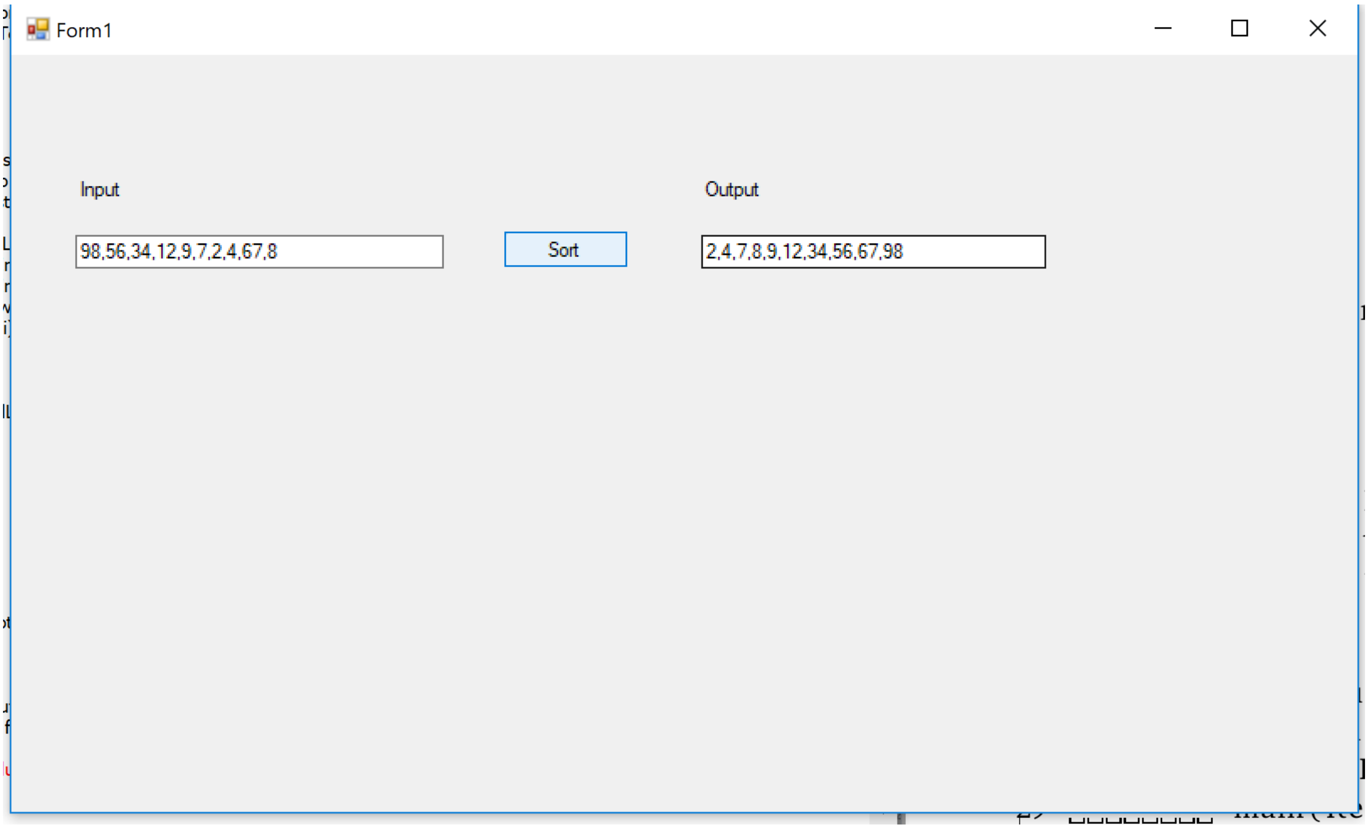
Assignment 3

A183 SQIT5013 KUMP A BUSINESS PROGRAMMING USING VISUAL TOOLS

Student Name:	Wong Jerng Foong
Metric Number:	824481
Lecturer:	Dr. Ruzelan Bin Khalid
Date:	July 24, 2019

1 Question 1-Selection Sort

1.1 Screenshot



98,56,34,12,9,7,2,4,67,8

2,4,7,8,9,12,34,56,67,98

Figure 1: Screenshot for Question 1

1.2 Steps

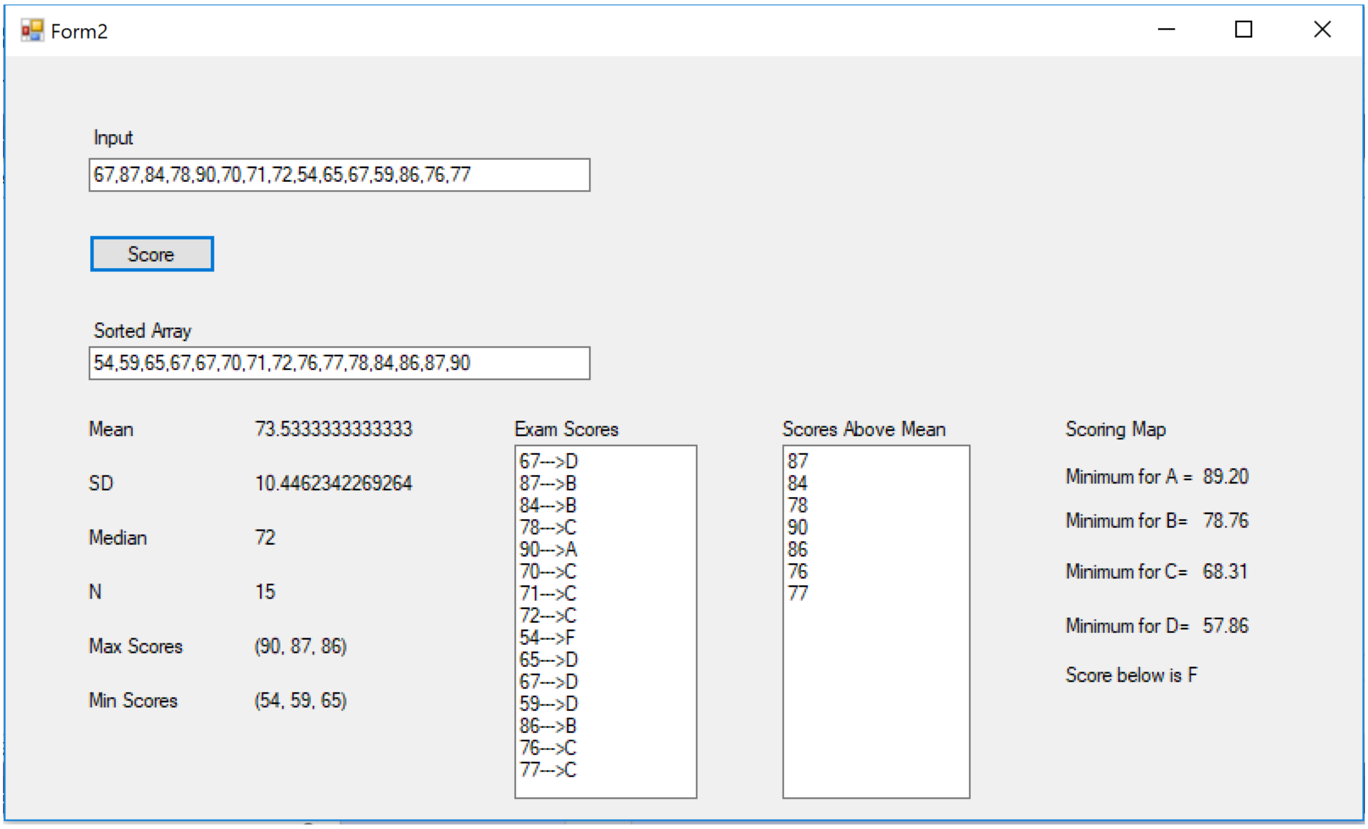
1. **ReadInput()**, gets the string from text input, and converts to integer array.
2. Do until all elements in array are processed:
   (a) **smallest()** will find the smallest value in array and return position of smallest element.
   (b) Based on position returned from **smallest()**,**swap()** will create a new list with 'swapped' elements between position 0 and position of smallest element.
   (c) Then, **newlist()** creates a new array, to be used for as searchspace, for the next pass. The new array will be a copy of previous array, but less the element at position 0.
   (d) At each pass, the element at position 0 (the smallest element of array) is added to a new array $intSortedList$.
3. At the end, the program will return the array $intSortedList$, which contains the elements of the original list, in sorted order.
4. Program is executed by button object btnSort, which executes the **sort1()** function.

1.3 Code

```
Public Class Form1
    Function smallest(list1() As Integer) As Integer
        ’inputs list
        ’returns position of smallest integer in array
        Dim intPosition As Integer
        Dim intSmallest As Integer = list1(0)
        For Each i As Integer In list1
            If i <= intSmallest Then
                intSmallest = i
                intPosition = Array.IndexOf(list1, i)
            End If

        Next

        Return intPosition


    End Function

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Sort.Click
        ’Button to execute main() as output
        Dim result As String = String.Join(",", sort1(ReadInput()))
        txtOutput.Text = result

    End Sub

    Function ReadInput() As Integer()
        ’Converts string text to integer array
        Dim s As String = txtInput.Text
        Dim numberstring As String() = s.Split(",")
        Dim intArray As Integer() = Array.ConvertAll(numberstring, Function(str) _
            Int32.Parse(Str))

        Return intArray

    End Function

    Function newlist(intOldList() As Integer) As _
        (intNewList1 As Integer(), smallestValue As Integer)
        ’input list, returns a tuple containing :(1)new list
        ’less Index zero element (2) smallest value
        Dim intNewList1(intOldList.Length - 2) As Integer
        For i = 1 To (intOldList.Length - 1)
            intNewList1(i - 1) = intOldList(i)
        Next
        Return (intNewList1, intOldList(0))
    End Function

    Function Swap(intPosition As Integer, list1() As Integer) As Integer()
        ’input position and list
        ’swaps smallest value to position zero,
        ’position zero value==>swapped integer position
        ’returns list with smallest at zero index position

        Dim intTempA As Integer = list1(0)
        Dim intTempB As Integer = list1(intPosition)
        Dim intList2 As Integer() = list1
        intList2(intPosition) = intTempA
        intList2(0) = intTempB

        Return intList2

    End Function

    Function sort1(intList1 As Integer()) As Integer()
        ’main function to run
        Dim intSortedList(intList1.Length - 1) As Integer

        For i = 0 To (intList1.Length - 1)
            Dim intA As Integer = smallest(intList1) ’returns position of smallest element
            Dim intB As Integer() = Swap(intA, intList1) ’create new list of swapped elements
            intList1 = newlist(intB).intNewList1 ’new list without zero index integer for next
                pass
            intSortedList(i) = newlist(intB).smallestValue ’ appends smallest to new list

        Next

        Return intSortedList


    End Function

End Class
```

2 Question 2-Exam Score Management Program

2.1 Screenshot



Input

67,87,84,78,90,70,71,72,54,65,67,59,86,76,77

Sorted Array

54,59,65,67,67,70,71,72,76,77,78,84,86,87,90

Mean	73.5333333333333
SD	10.4462342269264
Median	72
N	15
Max Scores	(90, 87, 86)
Min Scores	(54, 59, 65)

Exam Scores

```
67--->D
87--->B
84--->B
78--->C
90--->A
70--->C
71--->C
72--->C
54--->F
65--->D
67--->D
59--->D
86--->B
76--->C
77--->C
```

Scores Above Mean

```
87
84
78
90
86
76
77
```

Scoring Map

Minimum for A = 89.20

Minimum for B= 78.76

Minimum for C= 68.31

Minimum for D= 57.86

Score below is F

Figure 2: Screenshot for Question 2

2.2 Code

```
Public Class Form2
    Function smallest(list1() As Integer) As Integer
        'inputs list
        'returns position of smallest integer in array
        Dim intPosition As Integer
        Dim intSmallest As Integer = list1(0)
        For Each i As Integer In list1
            If i <= intSmallest Then
                intSmallest = i
                intPosition = Array.IndexOf(list1, i)
            End If

        Next

        Return intPosition


    End Function



    Function ReadInput() As Integer()
        'Converts string text to integer array
        'Reads text input and convert to array
        Dim s As String = txtInput.Text
        Dim numberstring As String() = s.Split(",")
        Dim intArray As Integer() = Array.ConvertAll(numberstring, Function(str) _
            Int32.Parse(str))

        Return intArray

    End Function

    Function newlist(intOldList() As Integer) As _
        (intNewList1 As Integer(), smallestValue As Integer)
        'input list, returns a tuple containing :(1)new list
        'less Index zero element (2) smallest value
        Dim intNewList1(intOldList.Length - 2) As Integer
        For i = 1 To (intOldList.Length - 1)
            intNewList1(i - 1) = intOldList(i)
        Next
        Return (intNewList1, intOldList(0))
    End Function

    Function Swap(intPosition As Integer, list1() As Integer) As Integer()
        'input position and list
        'swaps smallest value to position zero,
        'position zero value==>swapped integer position
        'returns list with smallest at zero index position

        Dim intTempA As Integer = list1(0)
        Dim intTempB As Integer = list1(intPosition)
        Dim intList2 As Integer() = list1
        intList2(intPosition) = intTempA
        intList2(0) = intTempB

        Return intList2

    End Function

    Function sort1(intList1 As Integer()) As Integer()
        'Return sorted list
        Dim intSortedList(intList1.Length - 1) As Integer

        For i = 0 To (intList1.Length - 1)
            Dim intA As Integer = smallest(intList1)
            Dim intB As Integer() = Swap(intA, intList1)
            intList1 = newlist(intB).intNewList1 'new list without zero index integer
            intSortedList(i) = newlist(intB).smallestValue
            ' appends smallest to new list

        Next

        Return intSortedList


    End Function

    Private Sub Button1_Click_1(sender As Object, e As EventArgs) Handles Button1.Click
        'Button to call and display all required outputs
        Dim result_mean As String = String.Join(",", Mean(ReadInput()))
        Dim result_sd As String = String.Join(",", Sd(ReadInput()))
        Dim result As String = String.Join(",", sort1(ReadInput()))
        Dim result_median As String = String.Join(",", Median(ReadInput()))
        Dim result_n As String = String.Join(",", (ReadInput().Length))
        Dim result_max As String = String.Join(",", Max(ReadInput()))
        Dim result_min As String = String.Join(",", Min(ReadInput()))
        Dim result_es As String = String.Join(",", ListScore(ReadInput()))
        Dim result_abovemean As String = String.Join(",", AboveMean(ReadInput()))
        txtOutput.Text = result
        lblMeanOutput.Text = result_mean 'display Mean
        lblSdOutput.Text = result_sd ' display Standard Deviation
        lblMedOutput.Text = result_median 'displays median value
        lblN.Text = result_n 'display number of values in array
        lblMax.Text = result_max 'display 3 max scores
        lblMin.Text = result_min 'display 3 min scores
        txtES.Text = result_es 'display exam score
        txtAboveMean.Text = result_abovemean 'display values above mean

        ''TEST
        'For checking minimum score of grades
        Dim m As Double = Mean(ReadInput())
        Dim StdDev As Double = Sd(ReadInput())
        ACondition.Text = Format(m + 1.5 * StdDev, "0.00")
        BCondition.Text = Format(m + 0.5 * StdDev, "0.00")
        CCondition.Text = Format(m - 0.5 * StdDev, "0.00")
        DCondition.Text = Format(m - 1.5 * StdDev, "0.00")




    End Sub
    Function Sum(intArray() As Integer) As Double
        'input array and output double
        'Return sum of array
        Dim dblSum As Double
        For Each i In intArray
            dblSum = i + dblSum
        Next
        Return dblSum

    End Function

    Function Mean(intArray() As Integer) As Double
        'input array and output double
        'Returns mean value from array input
        Dim dblSum As Double
        Dim dblMean As Double

        For Each i In intArray
            dblSum = i + dblSum
        Next

        dblMean = (dblSum / intArray.Length)

        Return dblMean
    End Function

    Function Sd(intArray() As Integer) As Double
        'input array, outputs double
        'Returns standard deviation value from array input

        Dim sum1 As Double
        Dim dblSd As Double

        For Each x In intArray
            sum1 = ((x - Mean(intArray)) ^ 2) + sum1
        Next
        dblSd = Math.Sqrt(sum1 / (intArray.Length - 1))

        Return dblSd
    End Function

    Function Median(intArray() As Integer) As Double
        Array.Sort(intArray)
        'Returns median value from array input

        If intArray.Length Mod 2 = 0 Then
            Return (intArray(intArray.Length \ 2 - 1) + intArray(intArray.Length \ 2)) / 2
        Else
            Return intArray(intArray.Length \ 2)
        End If
    End Function

    Function Min(intArray() As Integer) As (First As Integer, Second As Integer,
        Third As Integer)
        'input integer array and output tuple
        'Returns tuple values of 3 minimum values

        Dim intArrayMin As Integer()
        intArrayMin = sort1(intArray)

        Return (intArrayMin(0), intArrayMin(1), intArrayMin(2))


    End Function

    Function Max(intArray() As Integer) As (First As Integer, Second As Integer,
        Third As Integer)
        'input integer array and output tuple
        'Returns tuple of maximum three values

        Dim intArrayMax As Integer()
        intArrayMax = sort1(intArray)

        Return (intArrayMax(intArrayMax.Length - 1), intArrayMax(intArrayMax.Length - 2),
            intArrayMax(intArrayMax.Length - 3))

    End Function
    Function Score(ES As Integer, intArray As Integer()) As (Score As Integer, Grade As String)
        'Input double and return tuple of score and grade
        'Returns alphabetical score for numeric values input
        Dim m As Double = Mean(intArray)
        Dim s As Double = Sd(intArray)

        If ES >= m + 1.5 * s Then
            Return (ES, "A")
        ElseIf (ES >= m + 0.5 * s) And (ES < m + 1.5 * s) Then
            Return (ES, "B")
        ElseIf (ES >= m - 0.5 * s) And (ES < m + 0.5 * s) Then
            Return (ES, "C")
        ElseIf (ES >= m - 1.5 * s) And (ES < m - 0.5 * s) Then
            Return (ES, "D")
        Else
            Return (ES, "F")


        End If
    End Function

    Function ListScore(intArray As Integer()) As String
        ''input array, return string
        'A loop to score each value in array

        Dim string1 As String
        For Each i In intArray
            Dim ScoreGrade = Score(i, intArray)
            If string1 = "" Then ''to avoid blank carriage return for 1st pass
                string1 = string1 + ((ScoreGrade.Item1).ToString + "--->" + (ScoreGrade.Item2))

            Else string1 = string1 & vbCrLf & ((ScoreGrade.Item1).ToString + "--->" + (ScoreGrade
                .Item2))
            End If
        Next
        Return string1
    End Function

    Function AboveMean(intArray As Integer()) As String
        ''input array, return string
        'Return values which is above mean
        Dim m As Double = Mean(intArray)
        Dim string1 As String
        ''Dim ScoreGrade = Score(i, intArray)
        For Each i In intArray
            If i > m Then
                If string1 = "" Then ''to avoid blank carriage return for 1st pass

                    string1 = string1 + i.ToString

                Else string1 = string1 & vbCrLf + i.ToString

                End If
            End If

        Next
        Return string1
    End Function


End Class
```