 Computing Science

**Software Design and Development**

**Exemplification of standard algorithms required in Higher Computing Science in SQA Reference Language**

**Implementation of standard algorithms in VB 2010**

Contents

[Standard Algorithms – Input validation 3](#_Toc402466302)

[Validating integer input where lowerlimit =1 and upperlimit = 100 using a conditional While loop 3](#_Toc402466303)

[Standard Algorithms – Input validation 4](#_Toc402466304)

[Validating integer input where lowerlimit =1 and upperlimit = 100 using a conditional Repeat loop 4](#_Toc402466305)

[Standard Algorithms – Input validation 5](#_Toc402466306)

[Validating string input for YyNn using a conditional While loop 5](#_Toc402466307)

[Standard Algorithms – Input validation 6](#_Toc402466308)

[Validating string input for maximum length of 12 characters using a conditional While loop 6](#_Toc402466309)

[Standard Algorithms – Input validation 7](#_Toc402466310)

[Validating string input for maximum length of 12 characters using a conditional DO LOOP UNTIL loop and Boolean variable 7](#_Toc402466311)

[Standard Algorithms – Finding maximum 8](#_Toc402466312)

[Finding the maximum value from an array of 10 numbers 8](#_Toc402466313)

[Standard Algorithms – Finding minimum 9](#_Toc402466314)

[Finding the minimum value from an array of 10 numbers 9](#_Toc402466315)

[Standard Algorithms – Finding minimum and giving position in list 10](#_Toc402466316)

[Finding the minimum value from an array of 10 numbers and saying where in the list it occurs 10](#_Toc402466317)

[Standard Algorithms – Counting occurrences 11](#_Toc402466318)

[Counting the number of occurrences of a number in a list 11](#_Toc402466319)

[Standard Algorithms – Linear search 12](#_Toc402466320)

[Linear search of 10 element array giving either position of item or stating that item does not exist 12](#_Toc402466321)

All VB implementations need only a button to activate the code. No other objects are on the form.

# Standard Algorithms – Input validation

## Validating integer input where lowerlimit =1 and upperlimit = 100 using a conditional While loop

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| RECEIVE userInput FROM (INTEGER) KEYBOARDWHILE userInput < lowerLimit OR userInput > upperLimit DOSEND "Input must be between "& lowerLimit & " and " & upperLimit TO DISPLAY RECEIVE userInput FROM (INTEGER) KEYBOARDEND WHILE  | **Dim userinput, lowerlimit, upperlimit as Integer****lowerlimit = 1****upperlimit = 100****userinput=InputBox(“Please enter your number”)** **While userinput < 1 OR userinput > 100**  **MsgBox(“Input must be between 1 and 100”)** **userinput=InputBox(“Please enter your number”)****END While** |

# Standard Algorithms – Input validation

## Validating integer input where lowerlimit =1 and upperlimit = 100 using a conditional Repeat loop

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| REPEATRECEIVE userInput FROM (INTEGER) KEYBOARDIF userInput < lowerLimit OR userInput > upperLimit THEN SEND "Input must be between " & lowerLimit & " and "& upperLimit TO DISPLAY END IFUNTIL userInput >= lowerLimit AND userInput <= upperLimit  | **Dim userinput, lowerlimit, upperlimit as Integer****lowerlimit = 1****upperlimit = 100****Do** **userinput=InputBox(“Please enter your number”)** **IF userinput< lowerlimit OR userinput > upperlimit THEN** **MsgBox(“Input must be between 1 and 100”)****END IF****LOOP UNTIL userinput >= 1 AND userinput <=100** |

# Standard Algorithms – Input validation

## Validating string input for YyNn using a conditional While loop

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| RECEIVE userInput FROM (STRING) KEYBOARDWHILE userInput ∗= ["Y"] AND userInput ∗=["N"] DO SEND "Input must be Y or N " TO DISPLAYRECEIVE userInput FROM (STRING) KEYBOARDEND WHILE  | **Dim userinput as String****userinput =Ucase( InputBox(“Please enter Y or N”))****WHILE userinput <> “Y” AND userinput <> “N” DO** **MsgBox(“Input must be Y or N”)** **userinput =Ucase( InputBox(“Please enter Y or N”))****END WHILE** |

# Standard Algorithms – Input validation

## Validating string input for maximum length of 12 characters using a conditional While loop

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| RECEIVE userInput FROM (STRING) KEYBOARDWHILE LENGTH(userInput) >lengthlimit] DO SEND "Maximum 12 characters " TO DISPLAYRECEIVE userInput FROM (STRING) KEYBOARDEND WHILE  | **Dim userinput as String****userinput = InputBox(“Please enter string”)****WHILE LEN(userinput) >12**  **MsgBox(“Maximum 12 characters”)** **userinput = InputBox(“Please enter string”)****END WHILE** |

# Standard Algorithms – Input validation

## Validating string input for maximum length of 12 characters using a conditional DO LOOP UNTIL loop and Boolean variable

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| SET validInput TO false REPEATRECEIVE userInput FROM (STRING) KEYBOARDIF (length(userInput) > lengthLimit) THEN SET validInput TO TRUEELSESEND "Input must be less than " & lengthLimit & " characters" TO DISPLAY END IFUNTIL validInput = true   | **Dim userinput as String****Dim lengthlimit as Integer****Dim validinput as Boolean****lengthlimit =12****validinput = False****DO****userinput = InputBox(“Please enter string”)****IF LEN(userinput) > lengthlimit THEN** **validinput=True** **ELSE****MsgBox( "Input must be less than " & lengthLimit & " characters")****END IF****LOOP UNTIL** **validinput=True** |

# Standar**d Algorithms – Finding maximum**

## Finding the maximum value from an array of 10 numbers

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| SET maximumValue TO numbers[0] FOR counter FROM 1 TO 9 DOIF maximumValue < numbers[counter] THEN SET maximumValue TO numbers[counter]END IFEND FORSEND "The largest value was "& maximumValue TO DISPLAY | **Dim numbers = {2, 4, 6, 1, 12, 23, 6, 7, 10, 13}****Dim counter, maximumvalue as integer****maximumvalue = numbers(0)****FOR counter = 1 TO 9**  **IF maximumvalue<numbers(counter) THEN** **maximumvalue=numbers(counter)** **END IF****NEXT****MsgBox(“The maximum value in the list was “ & maximumvalue)** |

# Standard Algorithms – Finding minimum

## Finding the minimum value from an array of 10 numbers

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| SET minimumValue TO numbers[0] FOR counter FROM 1 TO 9 DOIF minimumValue > numbers[counter] THEN SET minimumValue TO numbers[counter]END IFEND FORSEND "The smallest value was "& minimumValue TO DISPLAY | **Dim numbers = {2, 4, 6, 1, 12, 23, 6, 7, 10, 13}****Dim counter, minimumvalue as integer****minimumvalue = numbers(0)****FOR counter = 1 TO 9**  **IF minimumvalue>numbers(counter) THEN** **minimumvalue=numbers(counter)** **END IF****NEXT****MsgBox(“The minimum value in the list was “ & minimumvalue)** |

# Standard Algorithms – Finding minimum and giving position in list

## Finding the minimum value from an array of 10 numbers and saying where in the list it occurs

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| SET foundAt TO 0SET minimumValue TO numbers[0]FOR index FROM 1 TO 9 DOIF minimumValue > numbers[index] THEN SET minimumValue TO numbers[index] SET foundAt TO indexEND IF END FORSEND "The smallest value was "& minimumValue & " at position "& foundAt & " in the list" TO DISPLAY  | **Dim numbers = {2, 4, 6, 1, 12, 23, 6, 7, 10, 13}****Dim counter, minimumvalue, foundAt as INTEGER****foundAt = 0****minimumvalue = numbers(0)****FOR counter = 1 TO 9**  **IF minimumvalue > numbers(counter) THEN** **minimumvalue=numbers(counter)** **foundAt = counter** **END IF****NEXT****MsgBox(“The minimum value in the list was “ & minimumvalue & “ at position “ & foundAt & “ in the list”)** |

# Standard Algorithms – Counting occurrences

## Counting the number of occurrences of a number in a list

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| RECEIVE itemToFind FROM (INTEGER) KEYBOARDSET numberFound TO 0FOREACH number FROM numbers DOIF number = itemToFind THENSET numberFound TO numberFound + 1END IFEND FOREACHSEND "There were " & numberFound & "occurrences of " & itemToFind & " in the list" TO DISPLAY | **Dim numbers = {2, 12, 6, 13, 12, 23, 6, 12, 10, 13}****Dim counter, itemToFind, numberFound as INTEGER****itemToFind = Inputbox(“What number are you searching for”)****numberFound = 0****FOR counter = 0 TO 9**  **IF itemToFind = numbers(counter) THEN** **numberFound =numberFound + 1** **END IF****NEXT****MsgBox(“There were “ & numberFound & “ occurrences of “ & itemToFind & “ in the list”)** |

# Standard Algorithms – Linear search

## Linear search of 10 element array giving either position of item or stating that item does not exist

|  |  |
| --- | --- |
| **Pseudocode** | **VB 2010 code** |
| SET found TO falseSET arraySize TO highestIndex RECEIVE itemToFind FROM (INTEGER) KEYBOARDREPEAT  IF numbers(counter) = itemToFind THEN SET found TO trueEND IFSET counter TO counter + 1 UNTIL counter > arraySize OR found = trueIF found = true THEN SEND itemToFind & " found at position" & counter-1 TO DISPLAY ELSE SEND "Item not found" TO DISPLAYEND IF | **Dim numbers={ 12, 3, 5, 13, 5, 67, 3, 34, 7, 61}** **Dim arraysize, counter, itemToFind as Integer****Dim found as Boolean****found = False****arraysize = 9****itemToFind=InputBox(“What number are you looking for?”)****DO** **If numbers(counter) = itemToFind THEN** **found = True** **END IF****counter = counter +1****LOOP UNTIL counter > arraysize OR found = True****IF found =True THEN** **MsgBox(itemToFind & “ found at position “ & counter +1)** **ELSE****MsgBox(“Item not found”)****END IF** |