Pre-Release M/J 2017 Solution

Computer science

Anas Othman, Abdulmonaim bensaber, & ahmad ayed

# TASK 1.1

# TASK 1.2

## MANUFACTURING

Sub Main()

Dim ProductIPrice As Integer

Dim Pdate As Date

Dim Edate As Date

Dim ProductW As Decimal

Dim FinalPrice As Decimal

Console.WriteLine("input product's initial price")

ProductIPrice = Console.ReadLine

Console.WriteLine("input product weight")

ProductW = Console.ReadLine

Console.WriteLine("input production date")

Pdate = Console.ReadLine

Console.WriteLine("input expiry date")

Edate = Console.ReadLine

FinalPrice = ProductW \* ProductIPrice

Console.Clear()

Console.WriteLine("your final product price is " & FinalPrice & " " &Pdate

" " & Edate )

Console.ReadLine()

End Sub

## COLLEGE / SCHOOL (AVERAGE)

Sub Main()

Dim Average As Integer

Dim Marks As Integer

Dim NumberOfStudents As Integer

Dim Total As Integer = 0

count = 1

Console.WriteLine("what is the number of students")

NumberOfStudents = Console.ReadLine

For count = 1 To NumberOfStudent

Console.WriteLine("input student marks")

Marks(count) = Console.ReadLine

Total = Total + Marks(count)

Next

Average = Total / NumberOfStudents

End Sub

# TASK 1.3

|  |  |
| --- | --- |
| Expression | Evaluates to |
| FlagA AND FlagB | FALSE |
| FlagB AND FlagC | FALSE |
| FLagB OR FlagC | TRUE |
| FlagA AND (FLagB OR FLagC) | TRUE |
| FlagB AND (NOT FLagB) | FALSE |
| (MyNum > 27) AND FLagC | FALSE |
| (MyNum >= 27) AND (FLagC = FALSE) | FALSE |

# TASK 2.1

## POST-CONDITION LOOP:

COUNT = 0

REPEAT

OUTPUT COUNT

UNTIL COUNT = 100

## PRE-CONDITION LOOP:

COUNT = 0

REPEAT

OUTPUT COUNT

WHILE COUNT <= 100

# TASK 2.2

DECLARE Start : Integer

DECLARE End : Integer

INPUT “Enter starting number: ” Start

INPUT “Enter ending number: ” End

REPEAT

If (Start Mod 3 = 0) Then

Output← ("Divisible by 3: " & Start)

End If

Start += 1

WHILE Start <= End

# TASK 3.1

DECLARE Word(10) : String

# TASK 3.2

FOR count IS 1 TO 10

Store “(EMPTY)” in Word(count)

END FOR

REPEAT

INPUT the CamelCase sentence

UNTIL first character of string is greater than or equal to “A” AND is less than or equal to “Z”

FOR j is equal to 1 TO 10

IF the length of the sentence is 0 THEN

EXIT the FOR loop

END IF

counter is equal to 0

IF first character of string is greater than or equal to “A” AND is less than or equal to “Z” THEN

FOR i is equal to 2 TO the length of the sentence

IF the next character starting from i in the sentence is greater than or equal to “a” AND is less than or equal to “z” THEN

INCREMENT counter by 1

ELSE

EXIT the FOR loop

ENDIF

END FOR

IF counter is greater than 0 THEN

(counter + 1) characters after the first character of the sentence is stored in Word(j)

The sentence is updated to a new one without the characters stored in Word(j)

ELSE

The first character of the sentence is stored in Word(j)

The sentence is updated to a new one without the first

character

END IF

ELSE

OUTPUT “String must only include letters”

END IF

END FOR

FOR count is equal to 1 TO 10

OUTPUT Word(count) and a new line

END FOR

## Identifier Table:

|  |  |  |
| --- | --- | --- |
| Variable | Data Type | Description |
| Word(10) | String | An array that stores the words from the sentence |
| CamelString | String | Stores the CamelCase sentence |
| count | Integer | A counter used in 2 FOR loops |
| j | Integer | A counter for a FOR loop |
| i | Integer | A counter for a FOR loop |
| counter | Integer | Stores the number of lower-cased letter in the sentence following an upper-cased letter |

# Task 3.3

DECLARE Word(10) : String

DECLARE CamelString : String

PROCEDURE EmptyArray

FOR count = 1 TO 10

  Word(count) ← “(EMPTY)”

END IF

END PROCEDURE

PROCEDURE Input

REPEAT

INPUT ← “Input sentence in CamelCase: ” CamelString

UNTIL LEFT(CamelString, 1) >= "A" AND LEFT(CamelString, 1) <= "Z"

END PROCEDURE

PROCEDURE Check

DECLARE counter : Integer

FOR j = 1 TO 10

  IF LENGTH(CamelString) = 0 THEN

        EXIT FOR

  END IF

  counter ← 0

IF LEFT(CamelString, 1) >= “A” AND LEFT(CamelString) <= “Z” THEN

  FOR i = 2 to LENGTH(CamelString)

        IF MID(CamelString, i, 1) >= “a” AND MID(CamelString,

i, 1) <= “z” THEN

counter = counter + 1

ELSE

EXIT For

END IF

END FOR

IF counter > 0 THEN

Word(j) ← MID(CamelString, 1, counter + 1)

CamelString ← MID(CamelString, counter + 2,

Len(CamelString) - counter - 1)

ElSE

Word(j) ← MID(CamelString, 1, 1)

CamelString ← MID(CamelString, 2, Len(CamelString) - 1)

END IF

ELSE

OUTPUT → “String must only include letters”

END IF

END FOR

END PROCEDURE

PROCEDURE Output

FOR count = 1 TO 10

OUTPUT → Word(count)

END FOR

END PROCEDURE

PROCEDURE Main

CALL EmptyArray

CALL Input

CALL Check

CALL Output

END PROCEDURE

# TASK 3.4

Dim Word(10) As String

Dim CamelString As String

Sub EmptyArray()

For count = 1 To 10

Word(count) = "(EMPTY)"

Next

End Sub

Sub Input()

Do Until Left(CamelString, 1) >= "A" And Left(CamelString, 1) <= "Z"

Console.Write("Input sentence in CamelCase: ")

CamelString = Console.ReadLine()

If Mid(CamelString, 1, 1) >= "a" And Mid(CamelString, 1, 1) <= "z" Then

Console.WriteLine("First letter must be capital")

End If

Loop

End Sub

Sub Check()

Dim counter As Integer

For j = 1 To 10

If Len(CamelString) = 0 Then

Exit For

End If

counter = 0

If Mid(CamelString, 1, 1) >= "A" And Mid(CamelString, 1, 1) <= "Z" Then

For i = 2 To Len(CamelString)

If Mid(CamelString, i, 1) >= "a" And Mid(CamelString, i, 1) <= "z" Then

counter = counter + 1

Else

Exit For

End If

Next

If counter > 0 Then

Word(j) = Mid(CamelString, 1, counter + 1)

CamelString = Mid(CamelString, counter + 2, (Len(CamelString) - counter – 1))

Else

Word(j) = Mid(CamelString, 1, 1)

CamelString = Mid(CamelString, 2, Len(CamelString) - 1)

End If

Else

Console.WriteLine("String must only include letters")

Exit For

End If

Next

End Sub

Sub Output()

For count = 1 To 10

Console.WriteLine(Word(count))

Next

End Sub

Sub Main()

Call EmptyArray()

Call Input()

Call Check()

Call Output()

End Sub

# TASK 3.5

For count = 1 To 10

Console.WriteLine(Word(count))

Next

# TASK 3.6

|  |  |
| --- | --- |
| Test Data | Justification |
| → “CamelString” | Valid data – Tests if data that should be accepted is so |
| → “caAmelCase” | Invalid data – Tests if data that should be rejected is so |
| → “AaAaAaAaAaAaAaAaAaAaAa”  → “Aa”  → “AaAaAaAaAaAaAaAaAaAaAaAa” | Boundary data – Tests the if the inputs just at the boundaries of what’s valid and what’s not ae accepted or rejected |
| → “1CamelCase”  → “Camel3Case”  → “&CamelCase”  → “Camel$Case” | Wrong format data – Tests if data of a different format is rejected as it should be |
| → (blank) | Absent data – Tests the behavior of the program when there’s no input |

# TASK 4

Dim Command As String

Dim b As String

Sub Main()

Console.WriteLine("1- Create and write file")

Console.WriteLine("2- Append file")

Console.WriteLine("3- Search")

Console.WriteLine("4- Append and search")

Console.Write("Enter command number: ")

Command = Console.ReadLine

If Command = "1" Then

Call Create()

ElseIf Command = "2" Then

Call Append()

ElseIf Command = "3" Then

Call Search()

ElseIf Command = "4" Then

Do

Console.Write("Write or search? Type 1 to write 2 to search and # to exit")

b = Console.ReadLine

If b = "1" Then

Call Append()

ElseIf b = "2" Then

Call Search()

End If

Loop Until b = "#"

End If

End Sub

Sub Search()

FileOpen(1, "record.txt", OpenMode.Input)

Dim SearchedNum As String

Dim MNum As String

Dim c As Integer

Do While True

Console.Write("Enter membership number: ")

SearchedNum = Console.ReadLine

If Len(SearchedNum) <> 6 Then

Console.WriteLine("Invalid input")

Else

Do Until EOF(1)

MNum = LineInput(1)

If Mid(MNum, 1, 6) = SearchedNum Then

Console.WriteLine(Mid(b, 8, 8))

c = 1

End If

Loop

End If

If c <> 1 Then

Console.WriteLine("Inputted membership number not found")

End If

FileClose(1)

Console.ReadLine()

Loop

End Sub

Sub Create()

FileOpen(1, "record.txt", OpenMode.Output)

Dim MembershipNum As Integer = 1

Dim VisitDate As String

Dim Line As String

Do While True

Console.Write("Enter the membership number; only integers are valid: ")

MembershipNum = Console.ReadLine

If Len(CStr(MembershipNum)) = 6 Then

Console.Write("Enter the date: ")

VisitDate = Console.ReadLine

If Len(CStr(VisitDate)) <> 8 Then

Console.WriteLine("Invalid date format")

Else

Line = CStr(MembershipNum) & " " & VisitDate

PrintLine(1, Line)

End If

Else

Console.WriteLine("Invalid membership number format")

End If

Loop

FileClose(1)

End Sub

Sub Append()

FileOpen(1, "record.txt", OpenMode.Append)

Dim MembershipNum As Integer = 1

Dim VisitDate As String

Dim Line As String

Do While True

Console.Write("Enter the membership number; only integers are valid: ")

MembershipNum = Console.ReadLine

If Len(CStr(MembershipNum)) = 6 Then

Console.Write("Enter the date: ")

VisitDate = Console.ReadLine

If Len(CStr(VisitDate)) <> 8 Then

Console.WriteLine("Invalid date format")

Else

Line = CStr(MembershipNum) & " " & VisitDate

PrintLine(1, Line)

End If

Else

If Len(CStr(MembershipNum)) <> 6 Then

Console.WriteLine("Invalid membership number format")

End If

End If

Loop

Console.ReadLine()

FileClose(1)

End Sub