Escuela Universitaria de Ingeniería Vitoria-Gasteiz
Ingeniaritzako Unibertsitate Eskola Vitoria-Gasteiz

Fundamentals of Computer Science

Exam – 9th June 2008

Resolution

eman ta zabal zazu

1. *(1 point)*

2. *(1.5 points)*

```
Sub cmd1_Click()
  Dim x As Integer, y As Integer
  If Not IsNumeric(txt1.Text) Or Not IsNumeric(txt2.Text) Then
    MsgBox "Operands must be numeric"
  Else
    x = Val(txt1.Text)
    y = Val(txt2.Text)
    If x = 0 Or y = 0 Then
      pct1.Print "Null"
    ElseIf x > 0 And y < 0 Or x < 0 And y > 0 Then
      pct1.Print "Negative"
    Else
      pct1.Print "Positive"
    End If
  End If
End Sub
```

Alternative (without **`ElseIf`***):*

```
    If x = 0 Or y = 0 Then
      pct1.Print "Null"
    Else
      If x > 0 And y < 0 Or x < 0 And y > 0 Then
        pct1.Print "Negative"
      Else
        pct1.Print "Positive"
      End If
    End If
```

3. *(3 points)*

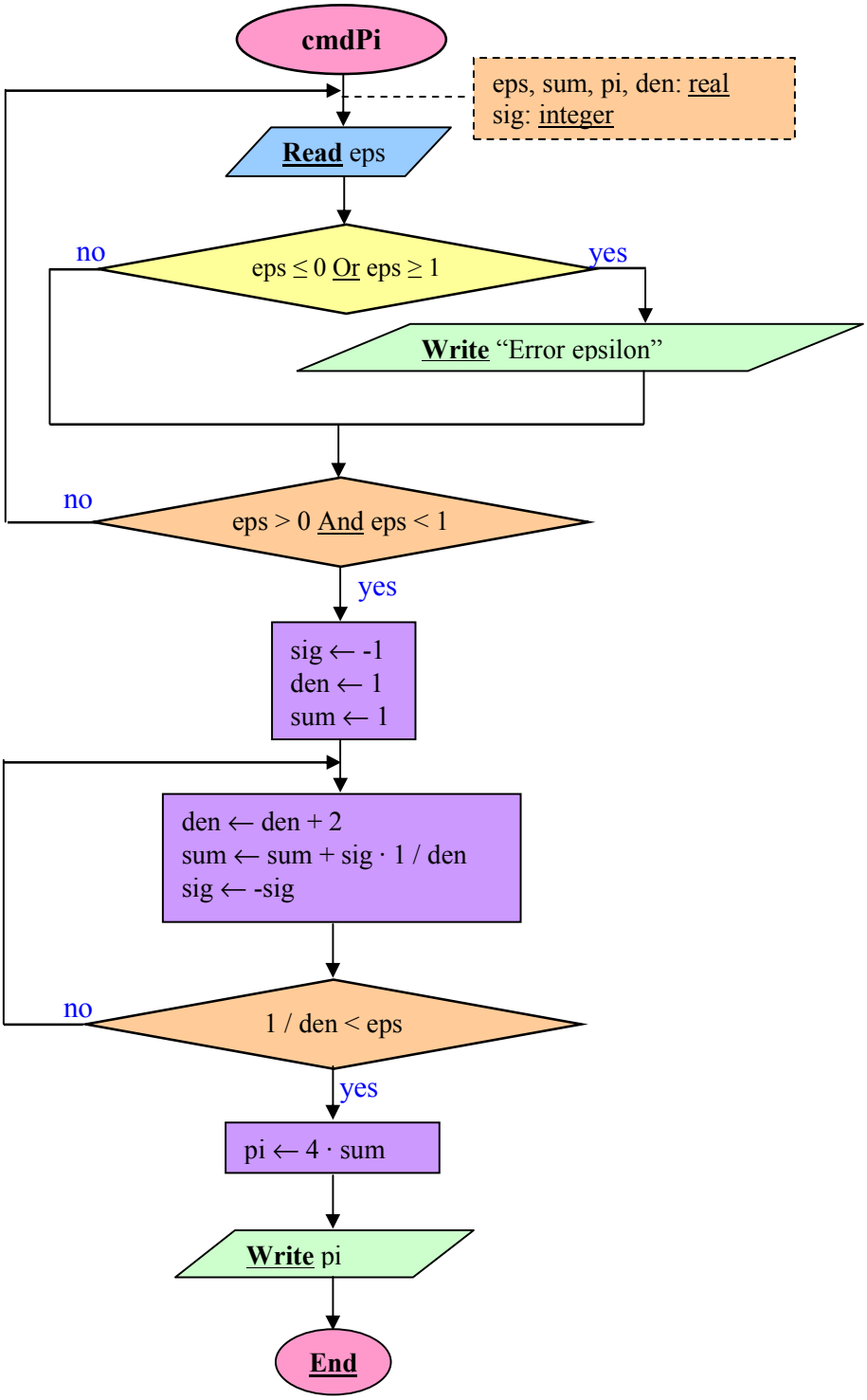
```
Function CheckPass (ByVal pass As String) As Integer
  Dim i As Integer, n As Integer
  Dim c As String
  Dim cd As Integer, cl As Integer, co As Integer
  n = Len(pass)
  If n < 4 Or n > 10 Then
    CheckPass = 1
  Else
    cd = 0 'Digits counter
    cl = 0 'Letters counter
    co = 0 'Others counter
    For i = 1 To n Step 1
      c = Mid(pass, i, 1)
      If c >= "0" And c <= "9" Then
        cd = cd + 1
      ElseIf c >= "a" And c <= "z" Or c >= "A" And c <= "Z" Then
        cl = cl + 1
      Else 'i.e. Not (c>="0" And c<="9" Or c>="a" And c<="z" Or c>="A" And c<="Z")
        co = co + 1
      End If
    Next i
    If cd = 0 Then
      CheckPass = 2
    ElseIf cl = 0 Then
      CheckPass = 3
    ElseIf co = 0 Then
      CheckPass = 4
    Else
      CheckPass = 0
    End If
  End If
End Function
```

Alternative (with Boolean variables instead of counters and first condition negated):

```
Function CheckPass (ByVal pass As String) As Integer
  Dim i As Integer, n As Integer
  Dim c As String
  Dim dig As Boolean, ltr As Boolean, oth As Boolean
  n = Len(pass)
  If n >= 4 And n <= 10 Then
    dig = False 'At least one digit
    ltr = False 'At least one letter
    oth = False 'At least one character that is neither a digit nor a letter
    For i = 1 To n Step 1
      c = Mid(pass, i, 1)
      If c >= "0" And c <= "9" Then
        dig = True
      ElseIf c >= "a" And c <= "z" Or c >= "A" And c <= "Z" Then
        ltr = True
      Else 'i.e. Not (c>="0" And c<="9" Or c>="a" And c<="z" Or c>="A" And c<="Z")
        oth = True
      End If
    Next i
    If Not dig Then
      CheckPass = 2
    ElseIf Not ltr Then
      CheckPass = 3
    ElseIf Not oth Then
      CheckPass = 4
    Else
      CheckPass = 0
  Else
    CheckPass = 1
  End If
End Function
```

Proposed alternative (not resolved): try defining (1) a function to count letters (2) a function to count digits. If there is at least one digit and one letter the sum of digits and letters must not be the total length of the string.

4. *(3.5 points)*
Flowchart:



VB program:

```
Sub cmdPi_Click()
  Dim eps As Double 'Epsilon
  Dim pi As Double
  Dim sum As Double 'Summation
  Dim den As Double 'Denominator
  Dim sig As Integer ‘ Sign
  Do
    eps = InputBox("Introduce epsilon")
    If eps <= 0 Or eps >= 1 Then
      MsgBox "The introduced value is not in between (0, 1)"
    End If
  Loop Until eps > 0 And eps < 1
  sig = -1
  den = 1
  sum = 1
  Do
    den = den + 2
    sum = sum + sig * 1 / den
    sig = sig * -1
  Loop Until 1 / den < eps
  pi = 4 * sum
  MsgBox pi
End Sub
```

Alternative (ellipses for repeated code – no flowchart provided):

```
Sub cmdPi_Click()
  Dim eps As Double 'Epsilon
  Dim pi As Double
  Dim sum As Double 'Summation
  Dim ter As Double 'Term
  Dim i As Integer ‘ Counter
... 'epsilon check equal
  sum = 1
  i = 1
  Do
    i = i + 1
    ter = 1 / (2 * i – 1)
    If i Mod 2 = 0 Then
      sum = sum - ter
    Else
      sum = sum + ter
    End If
  Loop Until ter < eps
  pi = 4 * sum
  MsgBox pi
End Sub
```

5. *(1 point)*

```
Function Desviation(ByVal n As Integer, v() As Double, ByVal m As Double) _
                    As Double
  Dim i As Integer
  Dim s As Double
  s = 0
  For i = 1 To n Step 1
    s = s + (v(i) - m) ^ 2
  Next i
  Desviation = Sqr(s / (n - 1))
End Function
```