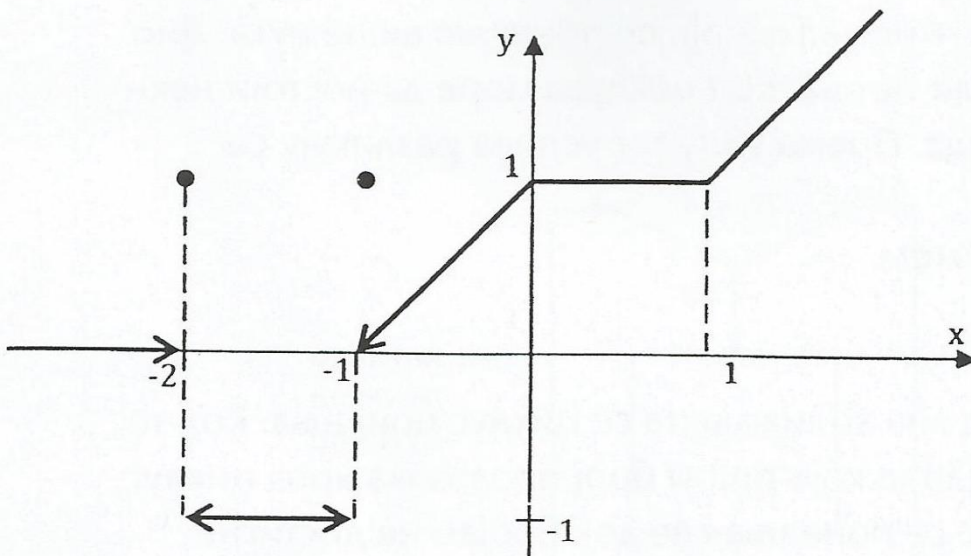


Priprema za 1 kolokvij

Zadatak 1

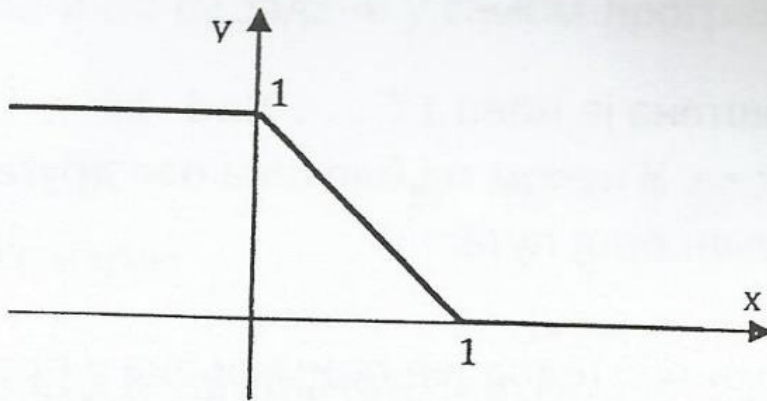
Саставити програм који за задане координате тачке (x,y) одређује квадрант којем тачка припада. Програм треба да открива и специјалне случајеве, кад је тачка у исходишту или на координатним осама



```

Select Case x
Case -1, -2, 0 To 1
    y = 1
Case Is < -2
    y = 0
Case -2 To -1
    y = -1
Case -1 To 0
    y = x + 1
Case Else
    y = x
End Select

```



```
Dim y As Single  
  
x = InputBox("")  
  
Select Case x  
    Case Is < 0  
        y = 1  
    Case 0 To 1  
        y = -x + 1  
    Case Else  
        y = 0  
End Select  
  
MsgBox (y)
```

Пример:

Примјер:

Саставити програм за израчунавање пресека двију правих из xu равни заданих у општем или имплицитном облику:

$$a_1x + b_1y + c_1 = 0$$

$$a_2x + b_2y + c_2 = 0$$

Рјешење:

Рјешење је тачка која лежи на обје праве тј. чије координате задовољавају једначину и једне и друге праве. До тих координата долази се израчунавањем двију непознатих x и y из двију датих једначина тј. рјешавањем предњег система линеарних једначина. Један од начина да се овај систем ријеша је Крамерово правило које до рјешења долази помоћу детерминанти. Према Крамеровом правилу рјешење предњег система гласи:

$$x = \frac{D_x}{D_s}, \quad y = \frac{D_y}{D_s}$$

гдје је:

$$D_s = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} - \text{детерминанта система,}$$

$$D_x = \begin{vmatrix} -c_1 & b_1 \\ -c_2 & b_2 \end{vmatrix} - \text{детерминанта од } x \text{ и}$$

$$D_y = \begin{vmatrix} a_1 & -c_1 \\ b_2 & -c_2 \end{vmatrix} - \text{детерминанта од } y.$$

Услов за постојање рјешења је да називник D_s у изразима за израчунавање x и y буде различит од нуле. То је знак да су праве паралелне. То је специфичан случај. Специфичан је и случај кад су и бројник и називник истовремено једнаки нули. То ће се десити кад су праве подударне. Програм не смије заказивати у тим специфичним ситуацијама, мора успјешно да се носи с њима. То се постиже уградњом дијелова кода који испитују евентуалну појаву специфичног случаја и одговарајућег одговора (поступања) када се он појави. *Квалитет програма у великој мјери зависи од тога колико је програмер успио да предвиди и адекватно одговори на све могуће специфичне ситуације.*

Слиједи комплетан програм постављеног рјешења.

```
Private Sub Form_Load()  
    Dim a1 As Double, b1 As Double, c1 As Double  
    Dim a2 As Double, b2 As Double, c2 As Double  
    Dim ds As Double, dx As Double, dy As Double  
    Dim x As Double, y As Double  
  
    a1 = 1.1: b1 = 1: c1 = 3  
    a2 = 2: b2 = 2: c2 = 6.5  
  
    ds = a1 * b2 - b1 * a2  
    dx = -c1 * b2 + c2 * b1  
    dy = -a1 * c2 + a2 * c1  
  
    If ds = 0 And dx = 0 Then  
        MsgBox ("Rjesenje je neodredjeno - prave su podudarne")  
    ElseIf ds = 0 Then  
        MsgBox ("Rjesenje je nemoguće - prave su paralelne")  
    Else  
        x = dx / ds  
        y = dy / ds  
        MsgBox ("x= " & x & Chr$(13) & "y= " & y)  
    End If  
End Sub
```


Примјер:

Саставити програм који задану ријеч испишује унатраг.

```
Private Sub Form_Load()  
    L% = Len(Rijec)  
  
    Kontra_Rijec$ = ""  
    For i% = L To 1 Step -1  
        Kontra_Rijec = Kontra_Rijec & Mid(Rijec, i, 1)  
    Next i  
  
    MsgBox (Kontra_Rijec)  
End Sub
```

Примјер:

Саставити програм који исписује слово по слово унесене ријечи све док се појави слово "s".

```
Private Sub Form_Load()  
    Rec$ = InputBox("")  
    i% = 1  
    Do  
        Slovo$ = Mid(Rec$, i, 1)  
        Debug.Print Slovo  
        i = i + 1  
    Loop Until Slovo = "s"  
End Sub
```


Примјер:

Саставити програм за израчунавање приближног рјешења нелинеарне једначине $f(x) = 0$ помоћу Њутновог метода тангенте.

Овај метод служи се тангентом на криву која представља график задане функције $y = f(x)$ да од неког предпостављеног почетног рјешења x_0 , дође до тачног рјешења. У почетној тачки (x_0, y_0) повлачи се тангента на криву и тражи њена пресјечна тачка са x осом. Тај пресјек је лако наћи јер је једначина тангенте линеарна:

$$y = y_0 + f'(x_0)(x - x_0)$$

па је за њен пресјек са x осом потребно ријешити систем:

$$y = y_0 + f'(x_0)(x - x_0)$$

$$y = 0$$

који се очигледно своди на:

$$0 = y_0 + f'(x_0)(x - x_0)$$

и рјешење добија у једном потезу

$$x_1 = x_0 - \frac{y_0}{f'(x_0)} = x_0 - \frac{f(x_0)}{f'(x_0)}$$

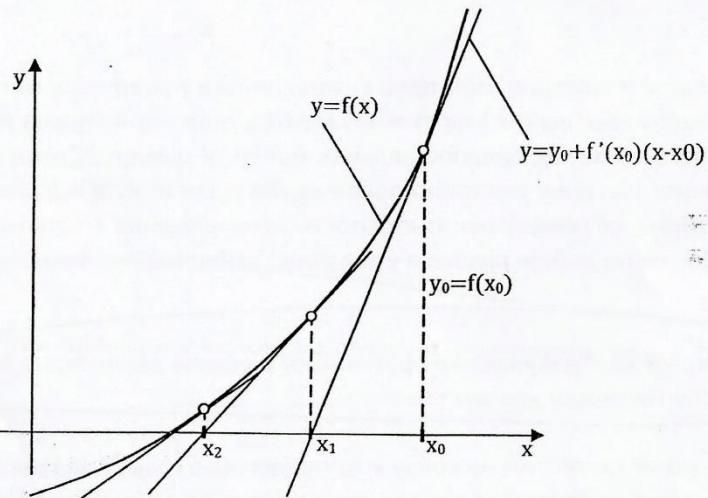
Добијена вриједност x_1 је ближа тачном рјешењу, а уколико није довољно близу поступак се понавља. Повлачи се тангента у новој тачки (x_1, y_1) , а њен пресјек са x осом

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

се још више приближава тачном рјешењу. Поступак се понавља док се не добије задовољавајућа тачност рјешења.

Поставља се питање како утврдити тачност приближног рјешења кад тачно рјешење незнамо. Постоје детаљне математичке методе како се та тачност може оцијенити. Ми ћемо тачност рјешења оцијенити по томе колико се у добијеној

тачки x_n крива приближила x оси. Према томе поступак ће се понављати све док се крива не приближи x оси на прихватљиво растојање. Ако за прихватљиво растојање усвојимо вриједност од 10^{-6} онда би излазни критеријум био

$$|y(x_n)| < 10^{-6}$$


Програмско рјешење дато је на сљедећем листингу. Оно не доноси нарочите новине које захтијевају додатно објашњење. Можда да истакнемо само то да рјешења добијена при поједином понављању циклуса, x_0, x_1, x_2, \dots , не представљају посебне варијабле. То чак не би било ни могуће јер не знамо колико ће таквих варијабли бити. Умјесто тога, свако поједино рјешење чува једна варијабла x чија се вриједност при сваком проласку кроз циклус мијења.

```
Private Sub Form_Load()
    Dim x As Double, y As Double, y_pr As Double

    x = InputBox("Unesite pocetno rjesenje:")

    Do
        y = 2 * x * x - 3 * x - 5
        y_pr = 4 * x - 3
        x = x - y / y_pr
    Loop Until Abs(y) < 0.000001

    MsgBox (x)
End Sub
```

Zadatak

Саставити програм који за задане координате тачке (x,y) одређује квадрант којем тачка припада. Програм треба да открива и специјалне случајеве, кад је тачка у исходишту или на координатним осама

Zadatak

Преpravити програм тако да се умјесто наредбе FOR ... NEXT употреби наредба DO ... LOOP

```
rijec=inputbox("Unesi rijec")
```

```
L=len(rijec)
```

```
n_upitnika = 0
```

```
For i = L to 1 step -1
```

```
If mid(rijec(i),1,1) = "?"
```

```
then n_upitnika = n_upitnika+1
```

```
Next i
```

Zadatak

Знаковна варијабла (string) **S** садржи презиме и име особе у наведеном редосљеду (раздвојене празнином). Саставити програм који мијења редосљед тако да се у варијабли **S** прво нађе име па онда презиме.

Zadatak

Саставити програм који израчунава површину круга описаног око правоугаоника са странама b и h .

Zadatak

Саставити BASIC програм који за задани датум мјесеца марта исписује име дана у седмици ако се зна да је први март пао у суботу.

Koju ће vriјednost poprimiti variјабла x по завршетку слједећег циклуса:

```
Dim x As Integer
```

```
x = 121
```

```
Do
```

```
x = x\5
```

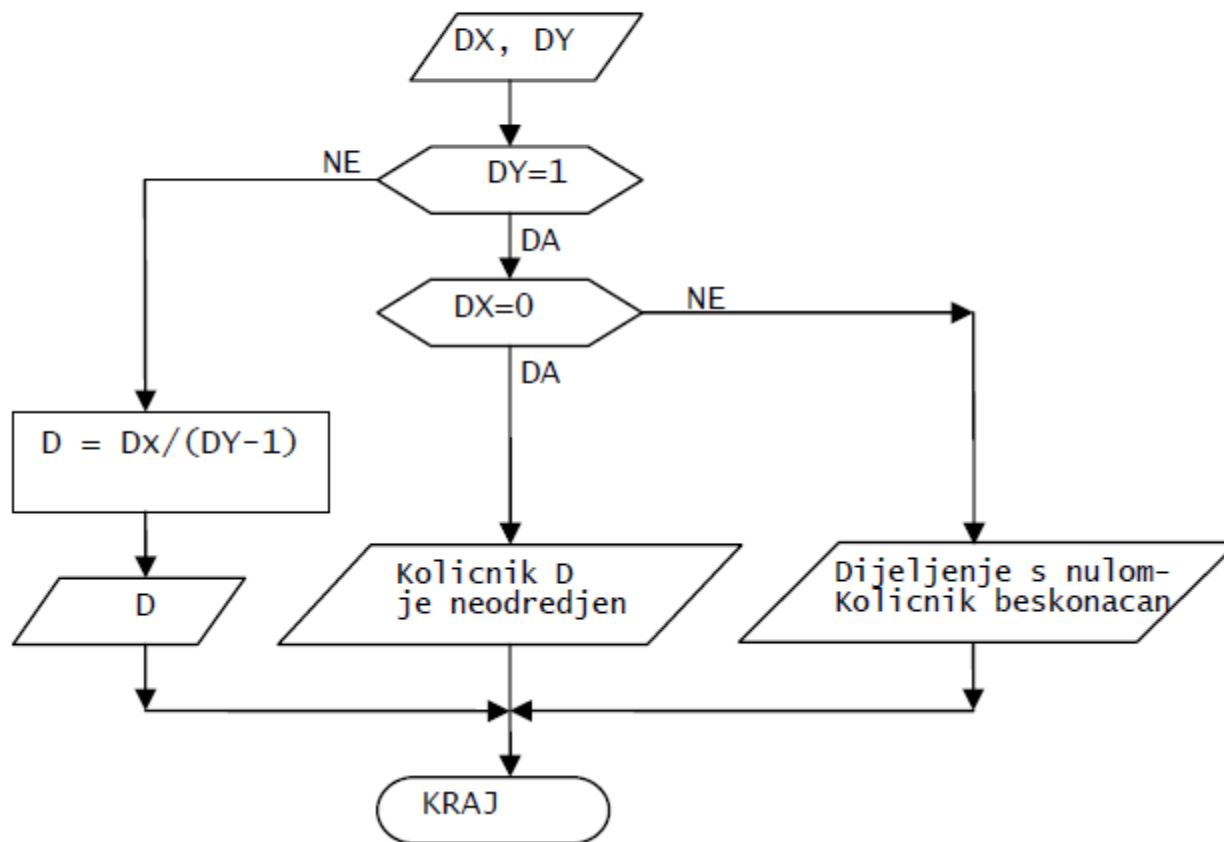
```
Loop While x>5
```

Zadatak

Саставити BASIC програм који за задани датум мјесеца Априла исписује име дана у седмици ако се зна да је први април пао у четвртак.

Zadatak

3. Програмирај алгоритам приказан на сљедећем дијаграму тока:



Zadatak

Koju ће vriјednost dobiti variјabla M у сљедећем BASIC програму?:

```
Dim x As Double
```

```
X = sin(2/3*atn(1))
```

Zadatak

Koju ће vrijednost poprimiti varijabla Rezultat po zavšetku sljedeћег дијела програма:

```
Dim Objekat As String, Rezultat As String
```

```
Dim L As Integer, i As Integer
```

```
Objekat = "Zakonodavstvo"
```

```
Rezultat = ""
```

```
L = Len(Objekat)
```

```
For i = 1 To L
```

```
If i Mod 2 Then
```

```
Rezultat = Rezultat & Mid(Objekat, i, 2)
```

```
End If
```

```
Next i
```