

# 1. uzdevums

Izveidot programmu, kura pēc funkcijas ievadītās argumenta x vērtības aprēķina tās vērtību.

$$f(x) = \begin{cases} (x+3)^2, & \text{ja } |x| < 3 \\ \log_2 x^2, & \text{ja } |x| > 10 \\ 81 - x^2, & \text{ja } 3 \leq x \leq 10 \end{cases}$$

## Kods:

```
# Programmas nosaukums: Funkcijas vērtību aprēķins
```

```
# 1. uzdevums (MPR5)
```

# Uzdevuma formulējums: Izveidot programmu, kura pēc funkcijas ievadītās argumenta x vērtības aprēķina tās vērtību.

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 1.1
```

```
import math # Lai izmantotu mat.log funkciju
```

```
x = float(input("Ievadiet argumenta x vērtību ==> ")) # Paprasīt lietotājam ievadīt x vērtību
```

```
if x<3 and x>-3 : # Ja x atrodas tādas robežas tad :
```

```
    a=(x+3)*(x+3)
```

```
    print("x = " + str(x) + " \nf(x) = " + (str(a))) # Izvada ekrāna vērtības
```

```
if x<-10 or x>10: # Ja x atrodas tādas robežas tad :
```

```
    a=math.log((x*x),2)
```

```
    print("x = " + str(x) + " \nf(x) = " + (str(a)))
```

```
if (x >= 3 and x <= 10) or (x <= -3 and x >=-10): # Ja x atrodas tādas robežas tad :
```

```
    a=81-x*x
```

```
    print("x = " + str(x) + " \nf(x) = " + (str(a)))
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Funkcijas vērtību aprēķins
# 1. uzdevums (MPR5)
# Uzdevuma formulējums: Izveidot programmu, kura pēc funkcijas ievadītās argumenta x vērtības aprēķina tās vērtību.
# Programmas autors: Vladislavs Babajins
# Versija 1.1

import math # Lai izmantotu mat.log funkciju

x = float(input("Ievadiet argumenta x vērtību ==> ")) # Paprasīt lietotājam ievadīt x vērtību

if x<3 and x>-3 : # Ja x atrodas tādas robežas tad :
    a=(x+3)*(x+3)
    print("x = " + str(x) + " \nf(x) = "+ (str(a))) # Izvada ekrāna vērtības

if x<-10 or x>10: # Ja x atrodas tādas robežas tad :
    a=math.log((x*x),2)
    print("x = " + str(x) + " \nf(x) = "+ (str(a)))

if (x >= 3 and x <= 10) or (x <= -3 and x>=-10): # Ja x atrodas tādas robežas tad :
    a=81-x*x
    print("x = " + str(x) + " \nf(x) = "+ (str(a)))
```

## Testa piemēri:

1)

```
Ievadiet argumenta x vērtību ==> 3
x = 3.0
f(x) = 72.0
```

2)

```
Ievadiet argumenta x vērtību ==> 10
x = 10.0
f(x) = -19.0
```

3)

```
Ievadiet argumenta x vērtību ==> 0
x = 0.0
f(x) = 9.0
```

4)

```
Ievadiet argumenta x vērtību ==> 12
x = 12.0
f(x) = 7.169925001442313
```

5)

```
Ievadiet argumenta x vērtību ==> 5
x = 5.0
f(x) = 56.0
```

6)

```
Ievadiet argumenta x vērtību ==> 100
x = 100.0
f(x) = 13.28771237954945
```

## 2. uzdevums (determinantu metode)

Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot determinantu metodi un ar koeficientu proporciju salīdzināšanas metodi.

### Kods:

```
# Programmas nosaukums: 2.uzdevums ar determinanta metodi.
```

```
# 2. uzdevums (MPR5)
```

```
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu
sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot
determinantu metodi un ar koeficientu proporciju salīdzināšanas metodi.
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 4.6
```

```
# Informācija lietotājam
```

```
print("2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera
formulas\n\nax + by = c\ndx + ey = f\n")
```

```
# a,b,c,d,e,f vērtību input
```

```
a = float(input("Ievadiet a vērtību ==> "))
```

```
b = float(input("Ievadiet b vērtību ==> "))
c = float(input("Ievadiet c vērtību ==> "))
d = float(input("Ievadiet d vērtību ==> "))
e = float(input("Ievadiet e vērtību ==> "))
f = float(input("Ievadiet f vērtību ==> "))
```

```
# Determinantu aprēķināšana
```

```
D=a*e-b*d
```

```
Dx=c*e-b*f
```

```
Dy=a*f-c*d
```

```
# Ja Sistēmas determinants ir 0 un (Dx != 0 vai Dy != 0 tad nav atrisinājumu)
```

```
if D==0 and (Dx !=0 or Dy !=0) :
```

```
    print("\n"+str(a)+"x + " + "(" + str(b) + ")" + "y" + " = "+str(c))
```

```
    print(str(d)+"x + " + "(" + str(e) + ")" + "y" + " = "+str(f))
```

```
    print("\nNav atrisinājumu")
```

```
# Ja visi determinanti (D == 0 un Dx == 0 un Dy == 0) tad ir bezgalīgi daudz atrisinājumu
```

```
if D==0 and Dx == 0 and Dy==0 :
```

```
    print("\n"+str(a)+"x + " + "(" + str(b) + ")" + "y" + " = "+str(c))
```

```
    print(str(d)+"x + " + "(" + str(e) + ")" + "y" + " = "+str(f))
```

```
    print("\nBezgalīgi daudz atrisinājumu")
```

```
# Ja sistēmas determinants nav 0, tad ir viens vienīgs atrisinājums
```

```
if D!=0:
```

$$x=Dx/D$$

$$y=Dy/D$$

```
print("\n"+str(a)+"x + " + "(" + str(b) + ")"+ "y"+"="+str(c))
```

```
print(str(d)+"x + " + "(" + str(e) + ")"+ "y"+"="+str(f))
```

```
print("\nx = "+str(x)+"\ny = "+str(y))
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: 2.uzdevums ar determinanta metodi.
# 2. uzdevums (99RS)
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot determinantu metodi un ar koeficientu proporciju
# Programmas autors: Vladislavs Babanins
# Versija 4.6

# Informācija lietotājam
print("2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmēra formulas\n\nax + by = c\nndx + ey = f\n")

# a,b,c,d,e,f vērtību input
a = float(input("Ievadiet a vērtību ==> "))
b = float(input("Ievadiet b vērtību ==> "))
c = float(input("Ievadiet c vērtību ==> "))
d = float(input("Ievadiet d vērtību ==> "))
e = float(input("Ievadiet e vērtību ==> "))
f = float(input("Ievadiet f vērtību ==> "))

# Determinantu aprēķināšana
D=a*e-b*d
Dx=c*e-b*f
Dy=a*f-c*d

# Ja Sistēmas determinants ir 0 un (Dx != 0 vai Dy != 0) tad nav atrisinājumu
if D==0 and (Dx !=0 or Dy !=0) :
    print("\n"+str(a)+"x + " + "(" + str(b) + ")"+ "y"+"="+str(c))
    print(str(d)+"x + " + "(" + str(e) + ")"+ "y"+"="+str(f))
    print("\nNav atrisinājumu")

# Ja visi determinanti (D == 0 un Dx == 0 un Dy == 0) tad ir bezgalīgi daudz atrisinājumu
if D==0 and Dx==0 and Dy==0 :
    print("\n"+str(a)+"x + " + "(" + str(b) + ")"+ "y"+"="+str(c))
    print(str(d)+"x + " + "(" + str(e) + ")"+ "y"+"="+str(f))
    print("\nBezgalīgi daudz atrisinājumu")

# Ja sistēmas determinants nav 0, tad ir viens vienīgs atrisinājums
if D!=0:
    x=Dx/D
    y=Dy/D
    print("\n"+str(a)+"x + " + "(" + str(b) + ")"+ "y"+"="+str(c))
    print(str(d)+"x + " + "(" + str(e) + ")"+ "y"+"="+str(f))
    print("\nx = "+str(x)+"\ny = "+str(y))
```

## Testa piemēri:

1)

```
2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmēra formulas

ax + by = c
dx + ey = f

Ievadiet a vērtību ==> 1
Ievadiet b vērtību ==> 2
Ievadiet c vērtību ==> 3
Ievadiet d vērtību ==> 4
Ievadiet e vērtību ==> 5
Ievadiet f vērtību ==> 6

1.0x + (2.0)y = 3.0
4.0x + (5.0)y = 6.0

x = -1.0
y = 2.0
```

2)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$ax + by = c$$

$$dx + ey = f$$

Ievadiet a vērtību ==> 2

Ievadiet b vērtību ==> 2

Ievadiet c vērtību ==> 2

Ievadiet d vērtību ==> 4

Ievadiet e vērtību ==> 4

Ievadiet f vērtību ==> 4

$$2.0x + (2.0)y = 2.0$$

$$4.0x + (4.0)y = 4.0$$

Bezgalīgi daudz atrisinājumu

3)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$ax + by = c$$

$$dx + ey = f$$

Ievadiet a vērtību ==> 2

Ievadiet b vērtību ==> 2

Ievadiet c vērtību ==> 2

Ievadiet d vērtību ==> 2

Ievadiet e vērtību ==> 2

Ievadiet f vērtību ==> 3

$$2.0x + (2.0)y = 2.0$$

$$2.0x + (2.0)y = 3.0$$

Nav atrisinājumu

4)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$ax + by = c$$

$$dx + ey = f$$

Ievadiet a vērtību ==> 2.4

Ievadiet b vērtību ==> 3.1313

Ievadiet c vērtību ==> 12.3

Ievadiet d vērtību ==> 2

Ievadiet e vērtību ==> 4

Ievadiet f vērtību ==> 6

$$2.4x + (3.1313)y = 12.3$$

$$2.0x + (4.0)y = 6.0$$

$$x = 9.112542697908554$$

$$y = -3.0562713489542768$$

5)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$ax + by = c$$

$$dx + ey = f$$

Ievadiet a vērtību ==> -2

Ievadiet b vērtību ==> -3

Ievadiet c vērtību ==> -5

Ievadiet d vērtību ==> -4

Ievadiet e vērtību ==> -2

Ievadiet f vērtību ==> -1

$$-2.0x + (-3.0)y = -5.0$$

$$-4.0x + (-2.0)y = -1.0$$

$$x = -0.875$$

$$y = 2.25$$

6)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$ax + by = c$$

$$dx + ey = f$$

Ievadiet a vērtību ==> -2

Ievadiet b vērtību ==> -2

Ievadiet c vērtību ==> -2

Ievadiet d vērtību ==> -4

Ievadiet e vērtību ==> -4

Ievadiet f vērtību ==> -3

$$-2.0x + (-2.0)y = -2.0$$

$$-4.0x + (-4.0)y = -3.0$$

Nav atrisinājumu

7)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$ax + by = c$$

$$dx + ey = f$$

Ievadiet a vērtību ==> 0

Ievadiet b vērtību ==> 0

Ievadiet c vērtību ==> 0

Ievadiet d vērtību ==> 0

Ievadiet e vērtību ==> 0

Ievadiet f vērtību ==> 0

$$0.0x + (0.0)y = 0.0$$

$$0.0x + (0.0)y = 0.0$$

Bezgalīgi daudz atrisinājumu

8)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$ax + by = c$$

$$dx + ey = f$$

Ievadiet a vērtību ==> 0

Ievadiet b vērtību ==> 1

Ievadiet c vērtību ==> 2

Ievadiet d vērtību ==> 0

Ievadiet e vērtību ==> 6

Ievadiet f vērtību ==> 2

$$0.0x + (1.0)y = 2.0$$

$$0.0x + (6.0)y = 2.0$$

Nav atrisinājumu



9)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$$

Ievadiet a vērtību ==> 0  
Ievadiet b vērtību ==> 1  
Ievadiet c vērtību ==> 0  
Ievadiet d vērtību ==> 2  
Ievadiet e vērtību ==> 3  
Ievadiet f vērtību ==> 4

$$\begin{aligned} 0.0x + (1.0)y &= 0.0 \\ 2.0x + (3.0)y &= 4.0 \end{aligned}$$

$$\begin{aligned} x &= 2.0 \\ y &= -0.0 \end{aligned}$$

Interessants piemērs:

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot Krāmera formulas

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$$

Ievadiet a vērtību ==> 0  
Ievadiet b vērtību ==> 0  
Ievadiet c vērtību ==> 0  
Ievadiet d vērtību ==> 0  
Ievadiet e vērtību ==> 0  
Ievadiet f vērtību ==> 1

$$\begin{aligned} 0.0x + (0.0)y &= 0.0 \\ 0.0x + (0.0)y &= 1.0 \end{aligned}$$

Bezgalīgi daudz atrisinājumu

Pareizā atbildē būtu “Nav atrisinājumu”. Bet tika vienots, ka lietotājs neievadīs 0 ka koeficientus, tāpēc es šo situāciju programmā neapskatīju (tas ir spec.gadījums).

Un Wolfram Alpha nevar dot atbildi:

NATURAL LANGUAGE

MATH INPUT

EXTENDED KEYBOARD

EXAMPLES

UPLOAD

RANDOM

Computational Inputs:

Assuming a system of two equations | Use [a system of three equations](#) or [more](#) instead

» equation 1:

» equation 2:

Compute

Try the following:

- Use different phrasing or notations
- Enter whole words instead of abbreviations
- Avoid mixing mathematical and other notations
- Check your spelling
- Give your input in English

## 2. uzdevums (ar proporciju salīdzināšanu metodi)

Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot determinantu metodi un ar koeficientu proporciju salīdzināšanas metodi.

```
# Programmas nosaukums: 2.uzdevums ar proporciju salīdzināšanu metodi.
```

```
# 2. uzdevums (MPR5)
```

```
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot determinantu metodi un ar koeficientu proporciju salīdzināšanas metodi.
```

```
# Programmas autors: Vladislavs Babaiņš
```

```
# Versija 2.4
```

```
# Informācija lietotājam
```

```
print("2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi\n\nax + by = c\ndx + ey = f\n")
```

```
# a,b,c,d,e,f vērtību input
```

```
a = float(input("Ievadiet a vērtību ==> "))
```

```
b = float(input("Ievadiet b vērtību ==> "))
```

```
c = float(input("Ievadiet c vērtību ==> "))
```

```
d = float(input("Ievadiet d vērtību ==> "))
```

```
e = float(input("Ievadiet e vērtību ==> "))
```

```
f = float(input("Ievadiet f vērtību ==> "))
```

```
# Determinantu aprēķināšana (Lai parādītu atrisinājumu, ja tads ir)
```

```
D=a*e-b*d
```

```
Dx=c*e-b*f
```

```
Dy=a*f-c*d
```

# Gadījums, kad nav atrisinājums

if a/d == b/e and b/e != c/f:

```
print("\n"+str(a)+"x + " + "(" + str(b) + ")" + "y" + " = "+str(c))
```

```
print(str(d)+"x + " + "(" + str(e) + ")" + "y" + " = "+str(f))
```

```
print("\nNav atrisinājumu")
```

# Gadījums, kad bezgalīgi daudz atrisinājumu

if a/d==b/e and b/e==c/f :

```
print("\n"+str(a)+"x + " + "(" + str(b) + ")" + "y" + " = "+str(c))
```

```
print(str(d)+"x + " + "(" + str(e) + ")" + "y" + " = "+str(f))
```

```
print("\nBezgalīgi daudz atrisinājumu")
```

# Ir viens atrisinājums

if a/d != b/e:

x=Dx/D

y=Dy/D

```
print("\n"+str(a)+"x + " + "(" + str(b) + ")" + "y" + " = "+str(c))
```

```
print(str(d)+"x + " + "(" + str(e) + ")" + "y" + " = "+str(f))
```

```
print("\nx = "+str(x)+"\ny = "+str(y))
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: 2. uzdevums ar proporciju salīdzināšanu metodi.
# 2. uzdevums (MP43)
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot determinantu metodi un ar koeficientu proporciju salīdzināšanas metodi.
# Programmas autori: Vladislavs Babajins
# Versija 2.4

# Informācija lietotājam
print("2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi\n\nax + by = c\n\nbx + dy = f\n\n")

# a,b,c,d,e,f vērtību input
a = float(input("Ievadiet a vērtību ==> "))
b = float(input("Ievadiet b vērtību ==> "))
c = float(input("Ievadiet c vērtību ==> "))
d = float(input("Ievadiet d vērtību ==> "))
e = float(input("Ievadiet e vērtību ==> "))
f = float(input("Ievadiet f vērtību ==> "))

# Determinantu aprēķināšana (lai parādītu atrisinājumu, ja tāds ir)
Dm=a*b-d
Dn=c*b-f
Dy=a*f-c*d

# Gadījums, kad nav atrisinājums
if a/d == b/e and b/e != c/f:
    print("\n" + str(a) + "x + " + str(b) + "y = " + str(c))
    print(str(d) + "x + " + str(e) + "y = " + str(f))
    print("\nNav atrisinājumu")

# Gadījums, kad bezgalīgi daudz atrisinājumu |
if a/d==b/e and b/e==c/f:
    print("\n" + str(a) + "x + " + str(b) + "y = " + str(c))
    print(str(d) + "x + " + str(e) + "y = " + str(f))
    print("\nBezgalīgi daudz atrisinājumu")

# Ir viens atrisinājums
if a/d != b/e:
    x=Dn/D
    y=Dy/D

    print("\n" + str(a) + "x + " + str(b) + "y = " + str(c))
    print(str(d) + "x + " + str(e) + "y = " + str(f))
    print("\nax + " + str(a) + "by = " + str(y))
```

## Testa piemēri:

1)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi

```
ax + by = c
dx + ey = f
```

```
Ievadiet a vērtību ==> 1
Ievadiet b vērtību ==> 2
Ievadiet c vērtību ==> 3
Ievadiet d vērtību ==> 4
Ievadiet e vērtību ==> 5
Ievadiet f vērtību ==> 6
```

```
1.0x + (2.0)y = 3.0
4.0x + (5.0)y = 6.0
```

```
x = -1.0
y = 2.0
```

2)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi

```
ax + by = c
dx + ey = f
```

```
Ievadiet a vērtību ==> 2
Ievadiet b vērtību ==> 2
Ievadiet c vērtību ==> 2
Ievadiet d vērtību ==> 2
Ievadiet e vērtību ==> 2
Ievadiet f vērtību ==> 2
```

```
2.0x + (2.0)y = 2.0
2.0x + (2.0)y = 2.0
```

Bezgalīgi daudz atrisinājumu

3)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$$

Ievadiet a vērtību ==> 2  
Ievadiet b vērtību ==> 2  
Ievadiet c vērtību ==> 1  
Ievadiet d vērtību ==> 2  
Ievadiet e vērtību ==> 2  
Ievadiet f vērtību ==> 4

$$\begin{aligned} 2.0x + (2.0)y &= 1.0 \\ 2.0x + (2.0)y &= 4.0 \end{aligned}$$

Nav atrisinājumu

4)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$$

Ievadiet a vērtību ==> -1  
Ievadiet b vērtību ==> -2  
Ievadiet c vērtību ==> -3  
Ievadiet d vērtību ==> -4  
Ievadiet e vērtību ==> -5  
Ievadiet f vērtību ==> -6

$$\begin{aligned} -1.0x + (-2.0)y &= -3.0 \\ -4.0x + (-5.0)y &= -6.0 \end{aligned}$$

$$\begin{aligned} x &= -1.0 \\ y &= 2.0 \end{aligned}$$

5)

2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi

$$\begin{aligned} ax + by &= c \\ dx + ey &= f \end{aligned}$$

Ievadiet a vērtību ==> 1  
Ievadiet b vērtību ==> 1  
Ievadiet c vērtību ==> 1  
Ievadiet d vērtību ==> 2  
Ievadiet e vērtību ==> 2  
Ievadiet f vērtību ==> 2

$$\begin{aligned} 1.0x + (1.0)y &= 1.0 \\ 2.0x + (2.0)y &= 2.0 \end{aligned}$$

Bezgalīgi daudz atrisinājumu

6)

```
2. kārtas lineāras vienādojumu sistēmas atrisinājums izmantojot proporciju salīdzināšanu metodi

ax + by = c
dx + ey = f

Ievadiet a vērtību ==> 14.14
Ievadiet b vērtību ==> 21.2
Ievadiet c vērtību ==> 151.122
Ievadiet d vērtību ==> 22
Ievadiet e vērtību ==> 5
Ievadiet f vērtību ==> 351

14.14x + (21.2)y = 151.122
22.0x + (5.0)y = 351.0

x = 16.89560272934041
y = -4.140652009097802
```

## PU1 2. uzdevums (determinantu metode)

Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot determinantu metodi un ar koeficientu proporciju salīdzināšanas metodi. Ar GUI.

### Kods:

```
# Programmas nosaukums: 2.uzdevums ar determinanta metodi ar GUI.
```

```
# PU1 (MPR5) 2.uzdevums
```

```
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt lineāras vienādojumu sistēmas koeficientu vērtības un paziņo tās atrisinājumu skaitu. Uzdevums jāatrisina, izmantojot determinantu metodi un ar koeficientu proporciju salīdzināšanas metodi. Ar GUI.
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 4.6
```

```
import math
```

```
from tkinter import * # Importējam tkinter moduli
```

```
root = Tk() # Tkinter (lai izmantotu to komandas)
```

```
root.geometry("230x185") # Loga izmēra definēšana
```

```
root.title("Lineāras vienādojumu sistēma") # Windows "loga" nosaukums
```

```
# Labels
```

```
Label_x1 = Label(root, text="x + ") #
```

```
Label_x1.place(x=50, y=50) #
```

```
Label_y1 = Label(root, text=" y = ") #
```

```
Label_y1.place(x=100, y=50) #
```

```
Label_x2 = Label(root, text="x + ") #
```

```
Label_x2.place(x=50, y=80) #
```

```
Label_y2 = Label(root, text=" y = ") #
```

```
Label_y2.place(x=100, y=80) #
```

```
# Tukšie "Label", lai varētu ievadīt un mainīt to saturošo informāciju
```

```
l1 = Label(root, text="")
```

```
l1.place(x=65, y=140)
```

```
# 1. vienādojums
```

```
Input_a=Entry(root, width=3)
```

```
Input_a.place(x=28,y=52)
```

```
Input_b=Entry(root, width=3)
```

```
Input_b.place(x=80,y=52)
```

```
Input_c=Entry(root, width=3)
```

```
Input_c.place(x=130,y=52)
```

```
# 2. vienādojums
```

```
Input_d=Entry(root, width=3)
```

```
Input_d.place(x=28,y=80)
```

```
Input_e=Entry(root, width=3)
```

```
Input_e.place(x=80,y=80)
```

```
Input_f=Entry(root, width=3)
```

```
Input_f.place(x=130,y=80)
```

```
def Result():
```

```
l1.config(text = "")
```

```
a=float(Input_a.get())
```

```
b=float(Input_b.get())
```

```
c=float(Input_c.get())
```

```
d=float(Input_d.get())
```

```
e=float(Input_e.get())
```

```
f=float(Input_f.get())
```



$D=a*e-b*d$

$Dx=c*e-b*f$

$Dy=a*f-c*d$

if  $D==0$  and ( $Dx \neq 0$  or  $Dy \neq 0$ ) :

l1.config(text = "Nav atrisinājumu")

if  $D==0$  and  $Dx == 0$  and  $Dy==0$  :

l1.config(text = "Bezgalīgi daudz atrisinājumu")

if  $D \neq 0$ :

$x=Dx/D$

$y=Dy/D$

l1.config(text = "x = "+str(x) + "\ny = " + str(y))

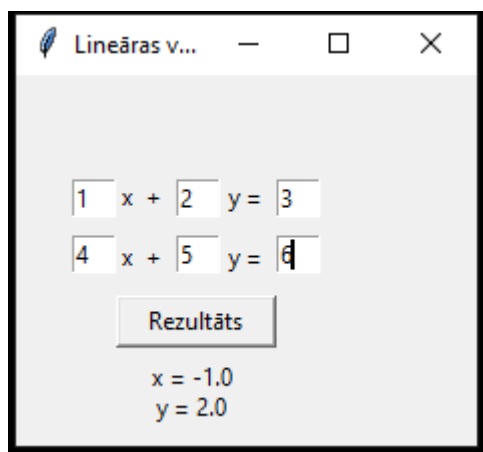
Submit\_button = Button(root, text="Rezultāts", command=Result) # Izmantojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta

Submit\_button.place(x=50, y=110, width=80) # Parādam, kur poga tiks attēlota

root.mainloop() # Obligāta rindiņa, lai logs butu redzāms visu laiku

## Testa piemēri:

1)



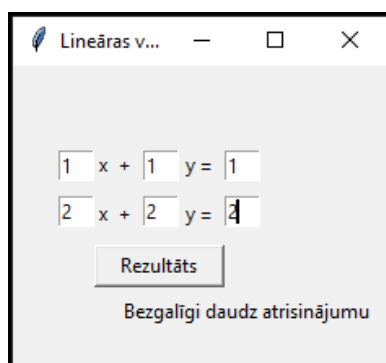
Lineāras v...

$$\begin{cases} 1x + 2y = 3 \\ 4x + 5y = 4 \end{cases}$$

Rezultāts

$$\begin{aligned} x &= -1.0 \\ y &= 2.0 \end{aligned}$$

2)



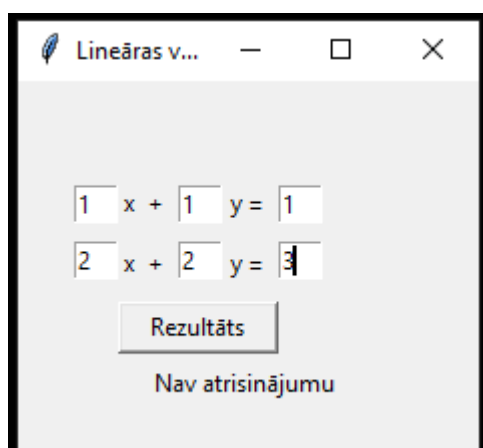
Lineāras v...

$$\begin{cases} 1x + 1y = 1 \\ 2x + 2y = 2 \end{cases}$$

Rezultāts

Bezgalīgi daudz atrisinājumu

3)



Lineāras v...

$$\begin{cases} 1x + 1y = 1 \\ 2x + 2y = 3 \end{cases}$$

Rezultāts

Nav atrisinājumu

4)

Lineāras v... — □ ×

$2.5x + 2y = 3.3$   
 $1x + 0y = 12$

Rezultāts

$x = 12.0$   
 $y = -13.35$

5)

Lineāras v... — □ ×

$12x + 22y = 33$   
 $6x + 2y = 4$

Rezultāts

$x = 0.2037037037037037$   
 $y = 1.3888888888888888$

6)

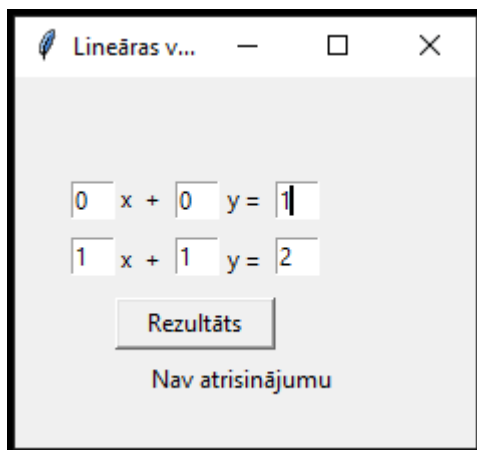
Lineāras v... — □ ×

$1x + 2y = -14$   
 $5x + 424y = 455$

Rezultāts

$x = -16.536231884057973$   
 $y = 1.2681159420289856$

7)



### 3. uzdevums

Izveidot programmu, kas pieprasa ievadīt kvadrātvienādojuma koeficientus un paziņo tā saknes (reālo un komplekso skaitļu kopā).

#### Kods:

```
# Programmas nosaukums: Kvadrātvienādojuma atrisināšana
```

```
# 3. uzdevums (MPR5)
```

```
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt kvadrātvienādojuma koeficientus un paziņo tā saknes (reālo un komplekso skaitļu kopā).
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 2.4
```

```
import math # Lai izmantotu math.sqrt()
```

```
print("Kvadrātvienādojumu atrisināšana")
```

```
a = float(input("Ievadiet a koeficientu ==> ")) # Input no lietotāja
```

```
b = float(input("Ievadiet b koeficientu ==> "))
```

```
c = float(input("Ievadiet c koeficientu ==> "))
```

```
# Nav kvadrātvienādojums pēc definīcijas
```

if a == 0 :

```
print("\n("+str(a)+")*x^2 + " + "(" + str(b) + ")" + "*x" + " + (" + str(c) + ") = 0")
print("\nTas nav kvadraatvienādojums")
```

else : # Citādi risinām kvadrātvienādojumu

d = b\*b - 4\*a\*c # Diskriminants

if d < 0 : # Tad ir komplēksa skaitļi

```
print("\n("+str(a)+")*x^2 + " + "(" + str(b) + ")" + "*x" + " + (" + str(c) + ") = 0")
print("\nKvadrātvienādojumam realu sakņu nav\nIr divas kompleksas saknes\n")
```

Re = -b

Im = abs(math.sqrt(-d) / (2 \* a))

print(f"x1 = {Re} + i\*{Im} \nx2 = {Re} - i\*{Im}")

elif d == 0 : # Tad divas vienādas saknes

x12 = -b / 2 / a

```
print("\n("+str(a)+")*x^2 + " + "(" + str(b) + ")" + "*x" + " + (" + str(c) + ") = 0")
print("\nKvadrātvienādojumam ir divas vienādas saknes")
print("x1 = x2 = " + str(x12))
```

else : # Divas atšķirīgas saknes

x1 = (-b + math.sqrt(d)) / 2 / a

$$x2 = (-b - \text{math.sqrt}(d)) / 2 / a$$

```
print("\n(" + str(a) + ") * x^2 + " + "(" + str(b) + ") * x + (" + str(c) + ") = 0")
```

```
print("\nKvadrātvienādojumam ir divas atšķirīgas saknes")
```

```
print("\nx1 = " + str(x1))
```

```
print("x2 = " + str(x2))
```

## Programmas kodu ekrānuuzņēmums:

```
# Programmas nosaukums: Kvadrātvienādojuma atrisināšana
# 3. uzdevums (MPR5)
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt kvadrātvienādojuma koeficientus un paziņo tā saknes (reālo un komplekso skaitļu kopā).
# Programmas autors: Vladislavs Babaiņš
# Versija 2.4

import math # Lai izmantotu math.sqrt()

print("Kvadrātvienādojumu atrisināšana")

a = float(input("Ievadiet a koeficientu ==> ")) # Input no lietotāja
b = float(input("Ievadiet b koeficientu ==> "))
c = float(input("Ievadiet c koeficientu ==> "))

# Nav kvadrātvienādojums pēc definīcijas
if a == 0 :

    print("\n(" + str(a) + ") * x^2 + " + "(" + str(b) + ") * x + (" + str(c) + ") = 0")
    print("\nTas nav kvadrātvienādojums")

else : # Citādi risinām kvadrātvienādojumu

    d = b*b - 4*a*c # Diskriminants

    if d < 0 : # Tad ir kompleksa skaitļi

        print("\n(" + str(a) + ") * x^2 + " + "(" + str(b) + ") * x + (" + str(c) + ") = 0")
        print("\nKvadrātvienādojumam reālu sakņu nav\nIr divas kompleksas saknes\n")

        Re = -b
        Im = abs(math.sqrt(-d)) / (2 * a)
        print(f"x1 = {Re} + i*{Im} \nx2 = {Re} - i*{Im}")

    elif d == 0 : # Tad divas vienas saknes

        x12 = -b / 2 / a

        print("\n(" + str(a) + ") * x^2 + " + "(" + str(b) + ") * x + (" + str(c) + ") = 0")
        print("\nKvadrātvienādojumam ir divas vienas saknes")
        print("x1 = x2 = " + str(x12))

    else : # Divas atšķirīgas saknes

        x1 = (-b + math.sqrt(d)) / 2 / a
        x2 = (-b - math.sqrt(d)) / 2 / a

        print("\n(" + str(a) + ") * x^2 + " + "(" + str(b) + ") * x + (" + str(c) + ") = 0")
        print("\nKvadrātvienādojumam ir divas atšķirīgas saknes")
        print("\nx1 = " + str(x1))
        print("x2 = " + str(x2))
```

## Testa piemēri:

1)

```
Kvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 1
Ievadiet b koeficientu ==> -2
Ievadiet c koeficientu ==> 1

(1.0)*x^2 + (-2.0)*x + (1.0) = 0

Kvadrātvienādojumam ir divas vienādas saknes
x1 = x2 = 1.0
```

2)

```
Kvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 3
Ievadiet b koeficientu ==> 2
Ievadiet c koeficientu ==> -3

(3.0)*x^2 + (2.0)*x + (-3.0) = 0

Kvadrātvienādojumam ir divas atšķirīgas saknes

x1 = 0.7207592200561265
x2 = -1.3874258867227933
```

3)

```
Kvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 1
Ievadiet b koeficientu ==> 0
Ievadiet c koeficientu ==> 9

(1.0)*x^2 + (0.0)*x + (9.0) = 0

Kvadrātvienādojumam realu sakņu nav
Ir divas kompleksas saknes

x1 = -0.0 + i*3.0
x2 = -0.0 - i*3.0
```

4)

```
Kvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> -2
Ievadiet b koeficientu ==> -3
Ievadiet c koeficientu ==> -5

(-2.0)*x^2 + (-3.0)*x + (-5.0) = 0

Kvadrātvienādojumam realu sakņu nav
Ir divas kompleksas saknes

x1 = 3.0 + i*1.3919410907075054
x2 = 3.0 - i*1.3919410907075054
```

5)

```
Kvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 2
Ievadiet b koeficientu ==> 3
Ievadiet c koeficientu ==> -2

(2.0)*x^2 + (3.0)*x + (-2.0) = 0

Kvadrātvienādojumam ir divas atšķirīgas saknes

x1 = 0.5
x2 = -2.0
```

6)

```
Kvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> -2.3
Ievadiet b koeficientu ==> 2.141
Ievadiet c koeficientu ==> -5

(-2.3)*x^2 + (2.141)*x + (-5.0) = 0

Kvadrātvienādojumam realu sakņu nav
Ir divas kompleksas saknes

x1 = -2.141 + i*1.3990294874005542
x2 = -2.141 - i*1.3990294874005542
```



## 4. uzdevums

Izveidot programmu, kas pieprasa ievadīt bikvadrātvienādojuma koeficientus un paziņo tā saknes reālo kopā. Uzzīmēt uzdevuma atrisināšanas blokshēmu.

### Kods:

```
# Programmas nosaukums: Bikvadrātvienādojuma atrisināšana
```

```
# 4. uzdevums (MPR5)
```

```
# Uzdevuma formulējums: Izveidot programmu, kas pieprasa ievadīt  
bikvadrātvienādojuma koeficientus un paziņo tā saknes reālo kopā. Uzzīmēt uzdevuma  
atsisināšanas blokshēmu.
```

```
# Programmas autors: Vladislavs Babaņins
```

```
# Versija 4.0
```

```
import math # Lai izmantotu math.sqrt()
```

```
print("Bikvadrātvienādojumu atrisināšana")
```

```
a = float(input("Ievadiet a koeficientu ==> "))
```

```
b = float(input("Ievadiet b koeficientu ==> "))
```

```
c = float(input("Ievadiet c koeficientu ==> "))
```

```
if a == 0 : # Pēc def. tas nav bikvadrātvienādojums ja a == 0
```

```
    print("\n(" + str(a) + ") * x^4 + " + "(" + str(b) + ") * x^2 + (" + str(c) + ") = 0")
```

```
    print("\nTas nav bikvadrātvienādojums")
```

```
else :
```

```
    d = b*b - 4*a*c # Diskriminants
```

if  $d < 0$  : # Risinām  $ax^4 + bx^2 + c = 0$ .  $x^2 = t$ ,  $at^2 + bt + c = 0$ , atrādam  $t_1, t_2$  un tad atrādam  $x_1, x_2$   $x = \pm\sqrt{t}$

```
print("\n(" + str(a) + ") * x^4 + " + "(" + str(b) + ") * x^2 + (" + str(c) + ") = 0\n")
```

```
print("\nBikvadrātvienādojumam nav reālas saknes")
```

else :

```
t1 = (-b + math.sqrt(d)) / 2 / a
```

```
t2 = (-b - math.sqrt(d)) / 2 / a
```

if  $t_1 \geq 0$  :

if  $t_2 \geq 0$ :

```
x1=math.sqrt(t1)
```

```
x2=-1*(math.sqrt(t1))
```

```
x3=math.sqrt(t2)
```

```
x4=-1*(math.sqrt(t2))
```

```
print("\n(" + str(a) + ") * x^4 + " + "(" + str(b) + ") * x^2 + (" + str(c) + ") = 0\n")
```

```
print("x1 = " + str(x1))
```

```
print("x2 = " + str(x2))
```

```
print("x3 = " + str(x3))
```

```
print("x4 = " + str(x4))
```

else:

```
x1= math.sqrt(t1)
```

```
x2 = -1*(math.sqrt(t1))
```

```
print("\n(" + str(a) + ") * x^4 + " + "(" + str(b) + ") * x^2 + (" + str(c) + ") = 0\n")
```

```
print("x1 = "+str(x1))
```

```
print("x2 = "+str(x2))
```

```
elif t2>=0:
```

```
    x1 = math.sqrt(t2)
```

```
    x2 = -1*(math.sqrt(t2))
```

```
    print("\n("+str(a)+")*x^4 + " + "(" + str(b) + ")" + "*x^2"+" + (" +str(c)+") = 0\n")
```

```
    print("x1 = "+str(x1))
```

```
    print("x2 = "+str(x2))
```

```
else:
```

```
    print("Sakņu nav")
```

### Testa piemēri:

1)

```
Bikvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 0
Ievadiet b koeficientu ==> 1
Ievadiet c koeficientu ==> 2

(0.0)*x^4 + (1.0)*x^2 + (2.0) = 0

Tas nav bikvadrātvienādojums
```

2)

```
Bikvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 1
Ievadiet b koeficientu ==> -3
Ievadiet c koeficientu ==> 2

(1.0)*x^4 + (-3.0)*x^2 + (2.0) = 0

x1 = 1.4142135623730951
x2 = -1.4142135623730951
x3 = 1.0
x4 = -1.0
```

3)

```
Bikvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 1
Ievadiet b koeficientu ==> 1
Ievadiet c koeficientu ==> -2

(1.0)*x^4 + (1.0)*x^2 + (-2.0) = 0

x1 = 1.0
x2 = -1.0
```

4)

```
Bikvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> 1
Ievadiet b koeficientu ==> 1
Ievadiet c koeficientu ==> 1

(1.0)*x^4 + (1.0)*x^2 + (1.0) = 0

Bikvadrātvienādojumam nav reālas saknes
```

5)

```
Bikvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> -0.5
Ievadiet b koeficientu ==> 1
Ievadiet c koeficientu ==> 1

(-0.5)*x^4 + (1.0)*x^2 + (1.0) = 0

x1 = 1.6528916502810695
x2 = -1.6528916502810695
```

6)

```
Bikvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> -9
Ievadiet b koeficientu ==> 1
Ievadiet c koeficientu ==> -1

(-9.0)*x^4 + (1.0)*x^2 + (-1.0) = 0

Bikvadrātvienādojumam nav reālas saknes
```

7)

```
Bikvadrātvienādojumu atrisināšana
Ievadiet a koeficientu ==> -10
Ievadiet b koeficientu ==> 10
Ievadiet c koeficientu ==> 0.6

(-10.0)*x^4 + (10.0)*x^2 + (0.6) = 0

x1 = 1.0279963211427374
x2 = -1.0279963211427374
```

8)

Bikvadrātvienādojumu atrisināšana

Ievadiet a koeficientu ==> -2

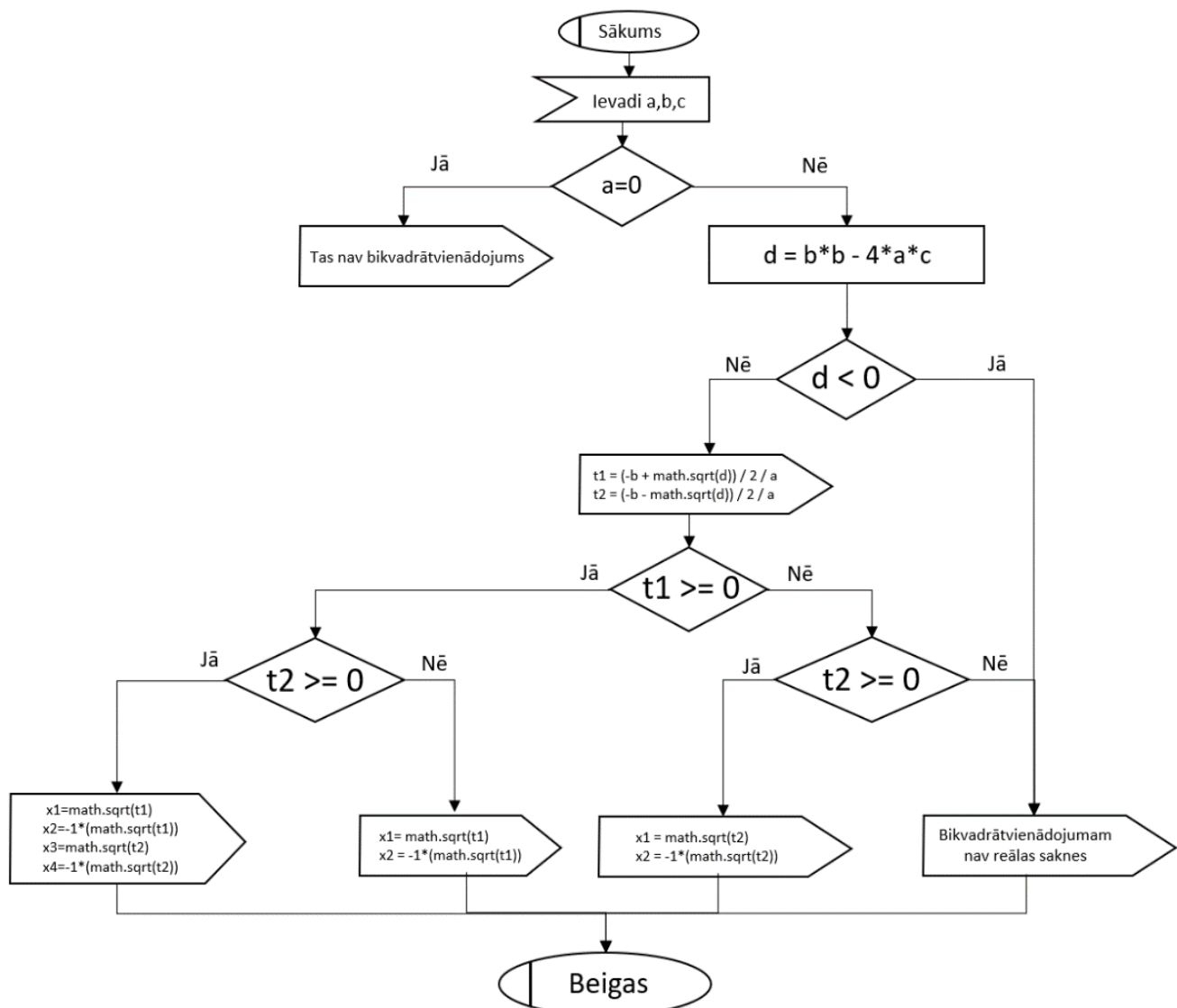
Ievadiet b koeficientu ==> -2

Ievadiet c koeficientu ==> -444

$$(-2.0)*x^4 + (-2.0)*x^2 + (-444.0) = 0$$

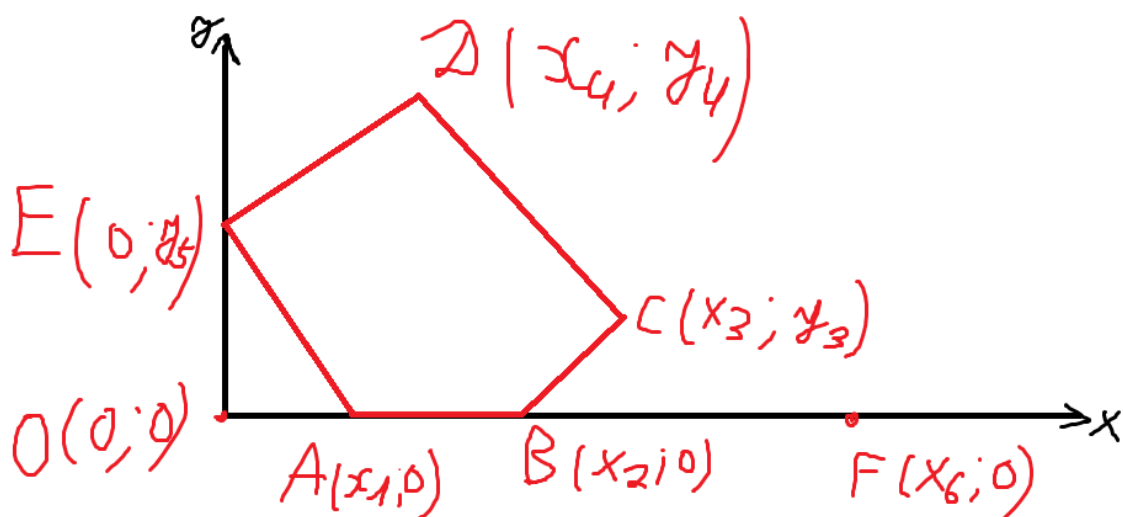
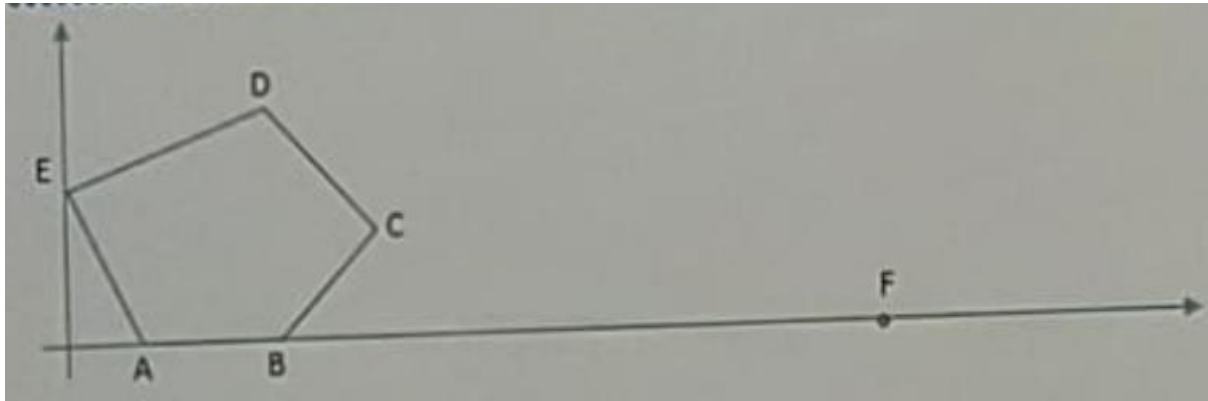
Bikvadrātvienādojumam nav reālas saknes

Blokshēma:



## 5. uzdevums

Dotais piecstūris bez slīdēšanas veļas x ass pozitīvajā virzienā. Noteikt, kura izliekta piecstūra mala vai virsotne uzkritis uz punktu F. Punktu koordinātas ievada lietotājs.



### Kods:

# Programmas nosaukums: Piecsturis bez slīdēšanas veļas x ass pozitīvajā virzienā...

# 5. uzdevums (MPR5)

# Uzdevuma formulējums: Dotais piecstūris bez slīdēšanas veļas x ass pozitīvajā virzienā. Noteikt, kura izliekta piecstūra mala vai virsotne uzkritis uz punktu F. Punktu koordinātas ievada lietotājs.

# Programmas autors: Vladislavs Babaņins

# Versija 1.0

```
import math
```

```
print("Izliekts piecsturis bez slidēšanas veļas x ass pozitīvajā virzienā.\nProgramma noteic  
kura izliekta piecstūra mala vai virsotne uzkritis uz punktu F.\nIevadiet koordinātas tikai  
1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!\n")
```

```
x1 = float(input("Ievadiet x1 ==> ")) # A punkts
```

```
x2 = float(input("Ievadiet x2 ==> ")) # B punkts
```

```
x3 = float(input("Ievadiet x3 ==> ")) # C punkts
```

```
y3 = float(input("Ievadiet y3 ==> "))
```

```
x4 = float(input("Ievadiet x4 ==> ")) # D punkts
```

```
y4 = float(input("Ievadiet y4 ==> "))
```

```
y5 = float(input("Ievadiet y5 ==> ")) # E punkts
```

```
x6 = float(input("Ievadiet x6 ==> ")) # F punkts
```

```
# Katras malas garums
```

```
#AB mala
```

```
AB=x2-x1
```

```
#BC mala
```

```
BC=math.sqrt((x3-x2)*(x3-x2)+(y3*y3))
```



#DC mala

$CD = \sqrt{(x_3 - x_4)^2 + (y_4 - y_3)^2}$

#DE mala

$DE = \sqrt{(y_4 - y_5)^2 + x_4^2}$

#EA mala

$EA = \sqrt{y_5^2 + x_1^2}$

# ----- #

#OA mala

$OA = x_1$

#OF mala

$OF = x_6$

# ----- #

#Perimets

$Perimeter = \text{int}(AB) + \text{int}(BC) + \text{int}(CD) + \text{int}(DE) + \text{int}(EA)$

if  $OF < OA$  :

    print("Piecsturis nekad neuzkritīs uz punktu F") # Ka  $OF < OA$  tad piecsturis neuzkritīs uz punktu F, jo piecsturis veļas x ass pozitīvajā virzienā.

else:

$z = (OF - OA)$  # z - ceļš kuru piecstūrim ir jāparverš

$u = \text{math.floor}(z / \text{Perimeter})$  # u - cik reizes ceļš ietilpst piecstūra perimetrā

$Atl = z - u * \text{Perimeter}$  # atl - atlikums

$AC = AB + BC$  # AC nogriežņa garums

$AD = AB + BC + CD$  # AD nogriežņa garums

$AE = AB + BC + CD + DE$  # AE nogriežņa garums

if  $Atl == 0$ :

print("F atrodas uz punkta A")

if  $Atl > 0$  and  $Atl < AB$ : # Atrodas šajos robežos

print("F atrodas uz AB nogriežņa")

if  $Atl == AB$ :

print("F atrodas uz punkta B")

if  $Atl > AB$  and  $Atl < AC$ :

print("F atrodas uz BC nogriežņa")

if  $Atl == AC$ :

print("F atrodas uz punkta C")

if  $Atl > AC$  and  $Atl < AD$ :

print("F atrodas uz CD nogriežņa")

if  $Atl == AD$  :

```
print("F atrodas uz punkta D")
```

if  $Atl > AD$  and  $Atl < AE$  :

```
print("F atrodas uz DE nogriežņa")
```

if  $Atl == AE$  :

```
print("F atrodas uz punkta E")
```

## Testa piemēri:

1)

```
Izliekts piecstūris bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 1  
Ievadiet x2 ==> 2  
Ievadiet x3 ==> 3  
Ievadiet y3 ==> 3  
Ievadiet x4 ==> 2  
Ievadiet y4 ==> 4  
Ievadiet y5 ==> 1  
Ievadiet x6 ==> 0  
Piecstūris nekad neuzkrītis uz punktu F
```

2)

```
Izliekts piecstūris bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 0  
Ievadiet x2 ==> 10  
Ievadiet x3 ==> 20  
Ievadiet y3 ==> 10  
Ievadiet x4 ==> 5  
Ievadiet y4 ==> 50  
Ievadiet y5 ==> 10  
Ievadiet x6 ==> 0  
F atrodas uz punkta A
```

3)

```
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 10  
Ievadiet x2 ==> 20  
Ievadiet x3 ==> 25  
Ievadiet y3 ==> 25  
Ievadiet x4 ==> 12  
Ievadiet y4 ==> 30  
Ievadiet y5 ==> 10  
Ievadiet x6 ==> 10000  
F atrodas uz CD nogriežņa
```

4)

```
Izliekts piecstūris bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 2  
Ievadiet x2 ==> 3  
Ievadiet x3 ==> 4  
Ievadiet y3 ==> 4  
Ievadiet x4 ==> 2  
Ievadiet y4 ==> 8  
Ievadiet y5 ==> 1  
Ievadiet x6 ==> 25  
F atrodas uz BC nogriežņa
```

5)

```
Izliekts piecstūris bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 1  
Ievadiet x2 ==> 2  
Ievadiet x3 ==> 3  
Ievadiet y3 ==> 4  
Ievadiet x4 ==> 5  
Ievadiet y4 ==> 6  
Ievadiet y5 ==> 7  
Ievadiet x6 ==> 8  
F atrodas uz CD nogriežņa
```

6)

```
Izliekts piecstūris bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 2  
Ievadiet x2 ==> 3  
Ievadiet x3 ==> 4  
Ievadiet y3 ==> 5  
Ievadiet x4 ==> 6  
Ievadiet y4 ==> 7  
Ievadiet y5 ==> 8  
Ievadiet x6 ==> 11  
F atrodas uz DE nogriežņa
```

7)

```
Izliekts piecsturis bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 1  
Ievadiet x2 ==> 2  
Ievadiet x3 ==> 3  
Ievadiet y3 ==> 4  
Ievadiet x4 ==> 5  
Ievadiet y4 ==> 6  
Ievadiet y5 ==> 7  
Ievadiet x6 ==> 20  
F atrodas uz punkta A
```

8)

```
Izliekts piecsturis bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 1  
Ievadiet x2 ==> 2  
Ievadiet x3 ==> 3  
Ievadiet y3 ==> 4  
Ievadiet x4 ==> 5  
Ievadiet y4 ==> 6  
Ievadiet y5 ==> 7  
Ievadiet x6 ==> 40  
F atrodas uz punkta B
```

9)

```
Izliekts piecsturis bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 2  
Ievadiet x2 ==> 3  
Ievadiet x3 ==> 3  
Ievadiet y3 ==> 4  
Ievadiet x4 ==> 6  
Ievadiet y4 ==> 6  
Ievadiet y5 ==> 2  
Ievadiet x6 ==> 66  
F atrodas uz DE nogriežņa
```

10)

```
Izliekts piecsturis bez slidēšanas veļas x ass pozitīvajā virzienā.  
Programma noteic kura izliekta piecstūra mala vai virsotne uzkrītis uz punktu F.  
Ievadiet koordinātas tikai 1.kvadrantā un tikai tādas koordinātas, lai veidotos izliekts piecstūris!  
  
Ievadiet x1 ==> 1  
Ievadiet x2 ==> 2  
Ievadiet x3 ==> 3  
Ievadiet y3 ==> 4  
Ievadiet x4 ==> 5  
Ievadiet y4 ==> 6  
Ievadiet y5 ==> 7  
Ievadiet x6 ==> 4546641641321  
F atrodas uz DE nogriežņa
```