

# 1. uzdevums

Izveidot programmu, kas atrisina vienādojumu sistēmu 
$$\begin{cases} ax^2 + by^2 + c = 0 \\ dx^2 + ey + f = 0 \end{cases}$$

## Kods:

```
# Programmas nosaukums: 1.uzdevums

# 1.uzdevums MPR6

# Uzdevuma formulējums: Izveidot programmu, kas atrisina vienādojumu sistēmu. ...
Koeficientus a, b, c, d, e ievada lietotājs

# Programmas autors: Vladislavs Babaņins

# Versija 1.4

import math

import sys

print("ax^2 + by^2 + c = 0\ndx^2 + ey + f = 0\n")

a = float(input("Ievadiet a ==> "))
b = float(input("Ievadiet b ==> "))

c = float(input("Ievadiet c ==> "))
d = float(input("Ievadiet d ==> "))

e = float(input("Ievadiet e ==> "))
f = float(input("Ievadiet f ==> "))

a_di = d*b
b_di = -1*a*e
c_di = d*c-a*f

D = (b_di)*(b_di) - 4*(a_di)*(c_di)
```

if  $D < 0$ :

```
print("Nav atrisinājumu")  
sys.exit(0)
```

elif  $d == 0$ :

```
print("Nav atrisinājumu")  
sys.exit(0)
```

elif  $a == 0$ :

```
print("Nav atrisinājumu")  
sys.exit(0)
```

else:

```
y1 = (-1*(b_di) + math.sqrt(D))/(2*a_di)  
y2 = (-1*(b_di) - math.sqrt(D))/(2*a_di)
```

```
k1 = (-1*c-b*y1*y1)/a
```

```
k2 = (-1*c-b*y2*y2)/a
```

if  $k2 > 0$  and  $k1 > 0$ :

```
x1 = math.sqrt(k1)
```

```
x2 = -1*math.sqrt(k1)
```

```
x3 = math.sqrt(k2)
```

```
x4 = -1*math.sqrt(k2)
```

```
print("x1 = " + str(x1) + " y1 = " + str(y1) + " x2 = " + str(x2) + " y2 = " + str(y2) )
```

```
print("x3 = " + str(x3) + " y2 = " + str(y2) + " x4 = " + str(x4) + " y2 = " + str(y2) )
```

else:

print("Nav atrisinājumu")

sys.exit(0)

## Testa piemēri:

1)

```
ax^2 + by^2 + c = 0
dx^2 + ey + f = 0

Ievadiet a ==> -4
Ievadiet b ==> 1
Ievadiet c ==> 1
Ievadiet d ==> -2
Ievadiet e ==> -2
Ievadiet f ==> -1
x1 = 1.5811388300841898 y1 = -3.0 x2 = -1.5811388300841898 y2 = -1.0
x3 = 0.7071067811865476 y2 = -1.0 x4 = -0.7071067811865476 y2 = -1.0
```

2)

```
ax^2 + by^2 + c = 0
dx^2 + ey + f = 0

Ievadiet a ==> 1
Ievadiet b ==> 1
Ievadiet c ==> 1
Ievadiet d ==> 1
Ievadiet e ==> 1
Ievadiet f ==> 1
Nav atrisinājumu
```

3)

$$\begin{aligned} ax^2 + by^2 + c &= 0 \\ dx^2 + ey + f &= 0 \end{aligned}$$

```
Ievadiet a ==> 1
Ievadiet b ==> 1
Ievadiet c ==> 1
Ievadiet d ==> 2
Ievadiet e ==> 2
Ievadiet f ==> -1
Nav atrisinājumu
```

## 2. uzdevums

Izveidot programmu, kas atrisina vienādojumu sistēmu  $\begin{cases} ax^2 + ay^2 + b = 0 \\ cxy + d = 0 \end{cases}$

Koeficientus a, b, c un d ievada lietotājs.

### Kods:

```
# Programmas nosaukums: 2.uzdevums
```

```
# 2.uzdevums MPR6
```

# Uzdevuma formulējums: Izveidot programmu, kas atrisina vienādojumu sistēmu. ...  
Koeficientus a, b, c un d ievada lietotājs.

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

```
# Versija 1.6
```

```
import math
```

```
print("ax^2 + ay^2 + b = 0\ncxy + d = 0\n")
```

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

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

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

```
if a==0 or c==0 :
```

```
    print("Specgadījums")
```

```
else:
```

```
    zemsakne1 = (-b/a) - ((2*d)/c)
```

```
    zemsakne2 = (-b/a) + ((2*d)/c)
```

```
if zemsakne1 < 0 or zemsakne2 < 0 :
```

```
    print("Nav risinājumu reālos skaitļos")
```

```
else:
```

```
    p1= math.sqrt(zemsakne1)
```

```
    p2= math.sqrt(zemsakne2)
```

```
    x = (-p1 + p2)/2
```

```
    y = (-p1 - p2)/2
```

```
if abs(x) == abs(y):
```

```
    print("x1 = " +str(x) + " y1 = " + str(y))
```

```
    print("x2 = " +str(-x) + " y2 = " + str(-y))
```

```
if abs(x) != abs(y):
```

```
    print("")
```

```
    print("x1 = " + str(x) + " y1 = " + str(y))
```

```
    print("x2 = " + str(y) + " y2 = " + str(x))
```

```
    print("x3 = " + str(-x) + " y3 = " + str(-y))
```

```
    print("x4 = " + str(-y) + " y4 = " + str(-x))
```

## Testa piemēri:

1)

```
ax^2 + ay^2 + b = 0
cxy + d = 0

Ievadiet a ==> 1
Ievadiet b ==> 1
Ievadiet c ==> 1
Ievadiet d ==> 1
Nav risinājumu reālos skaitļos
```

2)

```
ax^2 + ay^2 + b = 0
cxy + d = 0

Ievadiet a ==> 1
Ievadiet b ==> -1
Ievadiet c ==> -2
Ievadiet d ==> 1
x1 = -0.7071067811865476 y1 = -0.7071067811865476
x2 = 0.7071067811865476 y2 = 0.7071067811865476
```

3)

```
ax^2 + ay^2 + b = 0
cxy + d = 0

Ievadiet a ==> 0
Ievadiet b ==> 1
Ievadiet c ==> 2
Ievadiet d ==> 3
Specgadījums
```

### 3. uzdevums

Izveidot programmu, kas, pēc lietotāja ievadītajām trīs punktu A(x1, y1), B(x2, y2) un C(x3, y3) koordinātām, noskaidro un paziņo, vai šie trīs punkti atrodas uz vienas taisnes.

#### Kods:

```
# Programmas nosaukums: 3.uzdevums
```

```
# 3.uzdevums MPR6
```

# Uzdevuma formulējums: Izveidot programmu, kas, pēc lietotāja ievadītajām trīs punktu A(x1, y1), B(x2, y2) un C(x3, y3) koordinātām, noskaidro un paziņo, vai šie trīs punkti atrodas uz vienas taisnes.

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

```
# Versija 2.3
```

```
x1 = float(input("Ievadiet x1 ==> "))
```

```
y1 = float(input("Ievadiet y1 ==> "))
```

```
x2 = float(input("Ievadiet x2 ==> "))
```

```
y2 = float(input("Ievadiet y2 ==> "))
```

```
x = float(input("Ievadiet x3 ==> "))
```

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

```
a=(x-x2)*(y1-y2)
```

```
b=(y-y2)*(x1-x2)
```

```
if a==b:
```

```
    print("\nPunkti atrodas uz vienas taisnes")
```

```
if a!=b:
```

```
    print("\nPunkti nav uz vienas taisnes")
```

## Testa piemēri:

1)

```
Ievadiet x1 ==> 0
Ievadiet y1 ==> 0
Ievadiet x2 ==> 1
Ievadiet y2 ==> 1
Ievadiet x3 ==> 2
Ievadiet y3 ==> 2

Punkti atrodas uz vienas taisnes
```

2)

```
Ievadiet x1 ==> 0
Ievadiet y1 ==> 0
Ievadiet x2 ==> 2
Ievadiet y2 ==> 2
Ievadiet x3 ==> 4
Ievadiet y3 ==> 2

Punkti nav uz vienas taisnes
```

3)

```
Ievadiet x1 ==> 1
Ievadiet y1 ==> 2
Ievadiet x2 ==> 4
Ievadiet y2 ==> 8
Ievadiet x3 ==> -4
Ievadiet y3 ==> -8

Punkti atrodas uz vienas taisnes
```



## 4. uzdevums

Izveidot programmu, kas, pēc lietotāja secīgi ievadītajām četrstūra virsotņu A(x1, y1), B(x2, y2), C(x3, y3) un D(x4, y4) koordinātām, noskaidro un paziņo, vai šis četrstūris ir ieliekts vai izliekts.

### Kods:

```
# Programmas nosaukums: 4.uzdevums
```

```
# 4.uzdevums MPR6
```

# Uzdevuma formulējums: Izveidot programmu, kas, pēc lietotāja secīgi ievadītajām četrstūra virsotņu A(x1, y1), B(x2, y2), C(x3, y3) un D(x4, y4) koordinātām, noskaidro un paziņo, vai šis četrstūris ir ieliekts vai izliekts.

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

```
# Versija 2.4
```

```
# A
```

```
x1 = float(input("Ievadi X1 ==> "))
```

```
y1 = float(input("Ievadi Y1 ==> "))
```

```
# B
```

```
x2 = float(input("Ievadi X2 ==> "))
```

```
y2 = float(input("Ievadi Y2 ==> "))
```

```
# C
```

```
x3 = float(input("Ievadi X3 ==> "))
```

```
y3 = float(input("Ievadi Y3 ==> "))
```

```
# D
```

```
x4 = float(input("Ievadi X4 ==> "))
```

```
y4 = float(input("Ievadi Y4 ==> "))
```

```
#AC taisne
```

```
z1 = (x4 - x1)*(y3 - y1) - (y4 - y1)*(x3 - x1)
```

```
z2 = (x2 - x1)*(y3 - y1) - (y2 - y1)*(x3 - x1)
```

```
#BD taisne
```

```
z3 = (x1-x2)*(y4-y2)-(y1-y2)*(x4-x2)
```

```
z4 = (x3-x2)*(y4-y2)-(y3-y2)*(x4-x2)
```

```
if z1*z2 > 0 or z3*z4 > 0:
```

```
    print ("Izliekts")
```

```
else :
```

```
    print ("Ieliekts")
```

## Testa piemēri:

1)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 2
Ievadi Y2 ==> 6
Ievadi X3 ==> 1
Ievadi Y3 ==> 2
Ievadi X4 ==> 4
Ievadi Y4 ==> 4
Izliekts
```

2)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 4
Ievadi Y2 ==> 2
Ievadi X3 ==> 6
Ievadi Y3 ==> 2
Ievadi X4 ==> 6
Ievadi Y4 ==> 0
Ieliekts
```

3)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 2
Ievadi Y2 ==> -4
Ievadi X3 ==> 2
Ievadi Y3 ==> -2
Ievadi X4 ==> 6
Ievadi Y4 ==> -4
Izliekts
```

## 5. uzdevums

Izveidot programmu, kas, pēc lietotāja ievadītajam trijstūra virsotņu  $A(x_1, y_1)$ ,  $B(x_2, y_2)$  un  $C(x_3, y_3)$  un punkta  $D(x_4, y_4)$  koordinātām un paziņo, vai punkts  $D$  atrodas trijstūra  $ABC$  iekšpusē. Uzdevums jāatrisina izmantojot abas lekcijā aplūkotās metodes.

### Kods (1 metode):

# Programmas nosaukums: 5.uzdevums

# 5.uzdevums MPR6 (1 metode)

# Uzdevuma formulējums: Izveidot programmu, kas, pēc lietotāja ievadītajām trijstūra virsotņu A(x1, y1), B(x2, y2) un C(x3, y3), un punkta D(x4, y4) koordinātām, noskaidro un paziņo, vai punkts D atrodas trijstūra ABC iekšpusē. 1 metode.

# Programmas autors: Vladislavs Babajins

# Versija 1.3

```
import math
```

```
x1 = float(input("Ievadi X1 ==> "))
```

```
y1 = float(input("Ievadi Y1 ==> "))
```

```
x2 = float(input("Ievadi X2 ==> "))
```

```
y2 = float(input("Ievadi Y2 ==> "))
```

```
x3 = float(input("Ievadi X3 ==> "))
```

```
y3 = float(input("Ievadi Y3 ==> "))
```

```
x4 = float(input("Ievadi X4 ==> "))
```

```
y4 = float(input("Ievadi Y4 ==> "))
```

```
ab = math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))
```

```
bc = math.sqrt((x2-x3)*(x2-x3)+(y2-y3)*(y2-y3))
```

```
ac = math.sqrt((x1-x3)*(x1-x3)+(y1-y3)*(y1-y3))
```

```
ad = math.sqrt((x1-x4)*(x1-x4)+(y1-y4)*(y1-y4))
```

```
bd = math.sqrt((x2-x4)*(x2-x4)+(y2-y4)*(y2-y4))
```

```
cd = math.sqrt((x3-x4)*(x3-x4)+(y3-y4)*(y3-y4))
```

```
z1 = (x4 - x1) * (y2 - y1) - (x2 - x1) * (y4 - y1)
```

```
z2 = (x3 - x1) * (y2 - y1) - (x2 - x1) * (y3 - y1)
```

```
z3 = (x4 - x2) * (y3 - y2) - (x3 - x2) * (y4 - y2)
```

```
z4 = (x1 - x2) * (y3 - y2) - (x3 - x2) * (y1 - y2)
```

```
z5 = (x4 - x3) * (y1 - y3) - (x1 - x3) * (y4 - y3)
```

```
z6 = (x2 - x3) * (y1 - y3) - (x1 - x3) * (y2 - y3)
```

if (z1\*z2 > 0) and (z3\*z4 > 0) and (z5\*z6 > 0) :

```
print("Punkts D atrodas trijstūra iekšpusē")
```

else :

```
print("Punkts D neatrodas trijstūra iekšpusē")
```

### Testa piemēri:

1)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 4
Ievadi Y2 ==> 4
Ievadi X3 ==> 6
Ievadi Y3 ==> 0
Ievadi X4 ==> 4
Ievadi Y4 ==> 2
Punkts D atrodas trijstūra iekšpusē
```

2)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 2
Ievadi Y2 ==> 4
Ievadi X3 ==> 8
Ievadi Y3 ==> -4
Ievadi X4 ==> 2
Ievadi Y4 ==> 0
Punkts D atrodas trijstūra iekšpusē
```

3)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 2
Ievadi Y2 ==> 6
Ievadi X3 ==> 8
Ievadi Y3 ==> -4
Ievadi X4 ==> 0
Ievadi Y4 ==> -4
Punkts D neatrodas trijstūra iekšpusē
```

### Kods (2 metode):

```
# Programmas nosaukums: 5.uzdevums
```

```
# 5.uzdevums MPR6 (2 metode)
```

```
# Uzdevuma formulējums: Izveidot programmu, kas, pēc lietotāja ievadītajām trijstūra virsotņu A(x1, y1), B(x2, y2) un C(x3, y3), un punkta D(x4, y4) koordinātām, noskaidro un paziņo, vai punkts D atrodas trijstūra ABC iekšpusē. 2 metode.
```

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

```
# Versija 1.3
```

```
import math
```

```
x1 = float(input("Ievadi X1 ==> "))
```

```
y1 = float(input("Ievadi Y1 ==> "))
```

```
x2 = float(input("Ievadi X2 ==> "))
```

```
y2 = float(input("Ievadi Y2 ==> "))
```

```
x3 = float(input("Ievadi X3 ==> "))
```

```
y3 = float(input("Ievadi Y3 ==> "))
```

```
x4 = float(input("Ievadi X4 ==> "))
```

```
y4 = float(input("Ievadi Y4 ==> "))
```

```
ab = math.sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))
```

```
bc = math.sqrt((x2-x3)*(x2-x3)+(y2-y3)*(y2-y3))
```

```
ac = math.sqrt((x1-x3)*(x1-x3)+(y1-y3)*(y1-y3))
```

```
ad = math.sqrt((x1-x4)*(x1-x4)+(y1-y4)*(y1-y4))
```

```
bd = math.sqrt((x2-x4)*(x2-x4)+(y2-y4)*(y2-y4))
```

```
cd = math.sqrt((x3-x4)*(x3-x4)+(y3-y4)*(y3-y4))
```

```
z1 = (x4 - x1) * (y2 - y1) - (x2 - x1) * (y4 - y1)
```

```
z2 = (x3 - x1) * (y2 - y1) - (x2 - x1) * (y3 - y1)
```

```
z3 = (x4 - x2) * (y3 - y2) - (x3 - x2) * (y4 - y2)
```

```
z4 = (x1 - x2) * (y3 - y2) - (x3 - x2) * (y1 - y2)
```

```
z5 = (x4 - x3) * (y1 - y3) - (x1 - x3) * (y4 - y3)
```

```
z6 = (x2 - x3) * (y1 - y3) - (x1 - x3) * (y2 - y3)
```

```
if (z1*z2 > 0) and (z3*z4 > 0) and (z5*z6 > 0) :
```

```
    print("Punkts D atrodas trijstūra iekšpusē")
```

```
else :
```

```
print("Punkts D neatrodas trijstūra iekšpusē")
```

### Testa piemēri:

1)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 2
Ievadi Y2 ==> 6
Ievadi X3 ==> 8
Ievadi Y3 ==> -4
Ievadi X4 ==> 2
Ievadi Y4 ==> 2
Punkts D atrodas trijstūra iekšpusē
```

2)

```
Ievadi X1 ==> 0
Ievadi Y1 ==> 0
Ievadi X2 ==> 2
Ievadi Y2 ==> 2
Ievadi X3 ==> 4
Ievadi Y3 ==> 0
Ievadi X4 ==> 1000
Ievadi Y4 ==> 1000
Punkts D neatrodas trijstūra iekšpusē
```

3)



```
Ievadi X1 ==> -4  
Ievadi Y1 ==> -2  
Ievadi X2 ==> 0  
Ievadi Y2 ==> 4  
Ievadi X3 ==> 2  
Ievadi Y3 ==> 2  
Ievadi X4 ==> 0  
Ievadi Y4 ==> 2  
Punkts D atrodas trijstūra iekšpusē
```