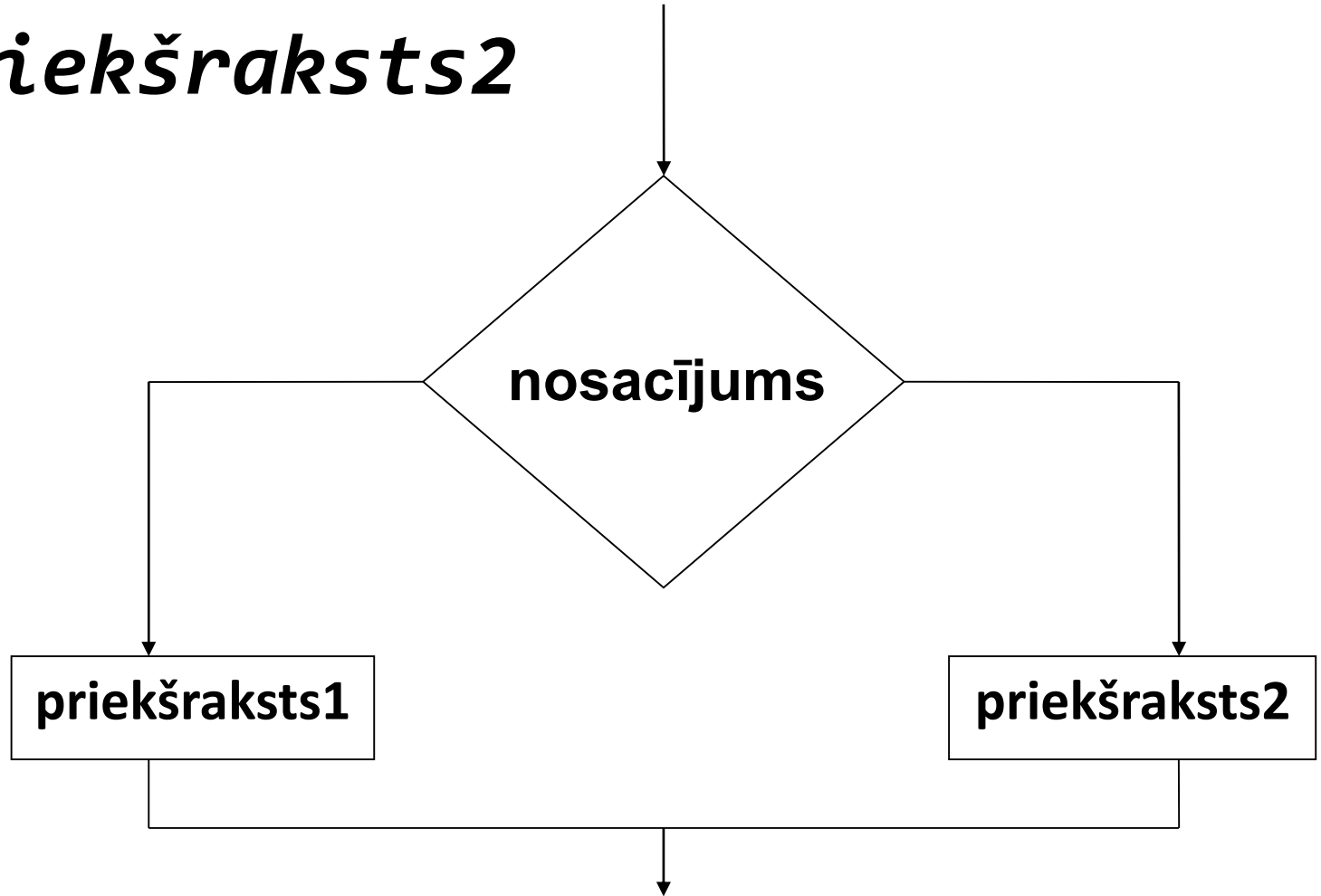


Zarošanās priekšraksts

```
if nosacījums:  
    priekšraksts1  
else:  
    priekšraksts2
```



```
# Divu skaitļu salīdzināšana
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
if b == a:
    print("Abi skaitļi ir vienādi")
```

```
# Var arī šādi, bet neiesaku!
```

```
#=====
```

```
# Divu skaitļu salīdzināšana
```

```
a = float(input("Ievadi pirmo skaitli ==>"))
```

```
b = float(input("Ievadi otro skaitli ==>"))
```

```
if b == a: print("Abi skaitļi ir vienādi")
```

Attiecības

- == vienāds
- != nevienāds
- > lielāks
- < mazāks
- >= lielāks vai vienāds
- <= mazāks vai vienāds

```
# Ievērot atkāpju lietošanu!!!  
# =====  
  
a = float(input("Ievadi pirmo skaitli ==>"))  
b = float(input("Ievadi otro skaitli ==>"))  
if b == a:  
print("Abi skaitļi ir vienādi")
```

Izvade:

File "C:\Users\Viesturs\Desktop\tests4.py", line 4, in ?
print("Abi skaitļi ir vienādi")

**builtins.IndentationError: expected an indented block
(tests4.py, line 4)**

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
if b == a:
    print("Abi skaitļi ir vienādi")
if b != a:
    print("Abi skaitļi nav vienādi")
```

```
# Ievēro atkāpju lietošanu!!!
```

```
#=====
```

```
a = float(input("Ievadi pirmo skaitli ==>"))
```

```
b = float(input("Ievadi otro skaitli ==>"))
```

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

```
    print("Abi skaitļi ir vienādi")
```

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

```
    print("Abi skaitļi nav vienādi")
```



```
# Divu skaitļu salīdzināšana
```

```
a = float(input("Ievadi pirmo skaitli ==>"))
```

```
b = float(input("Ievadi otro skaitli ==>"))
```

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

```
    print("Abi skaitļi ir vienādi")
```

```
else:
```

```
    print("Abi skaitļi nav vienādi")
```

```
# Var arī šādi, bet neiesaku!
```

```
#=====
```

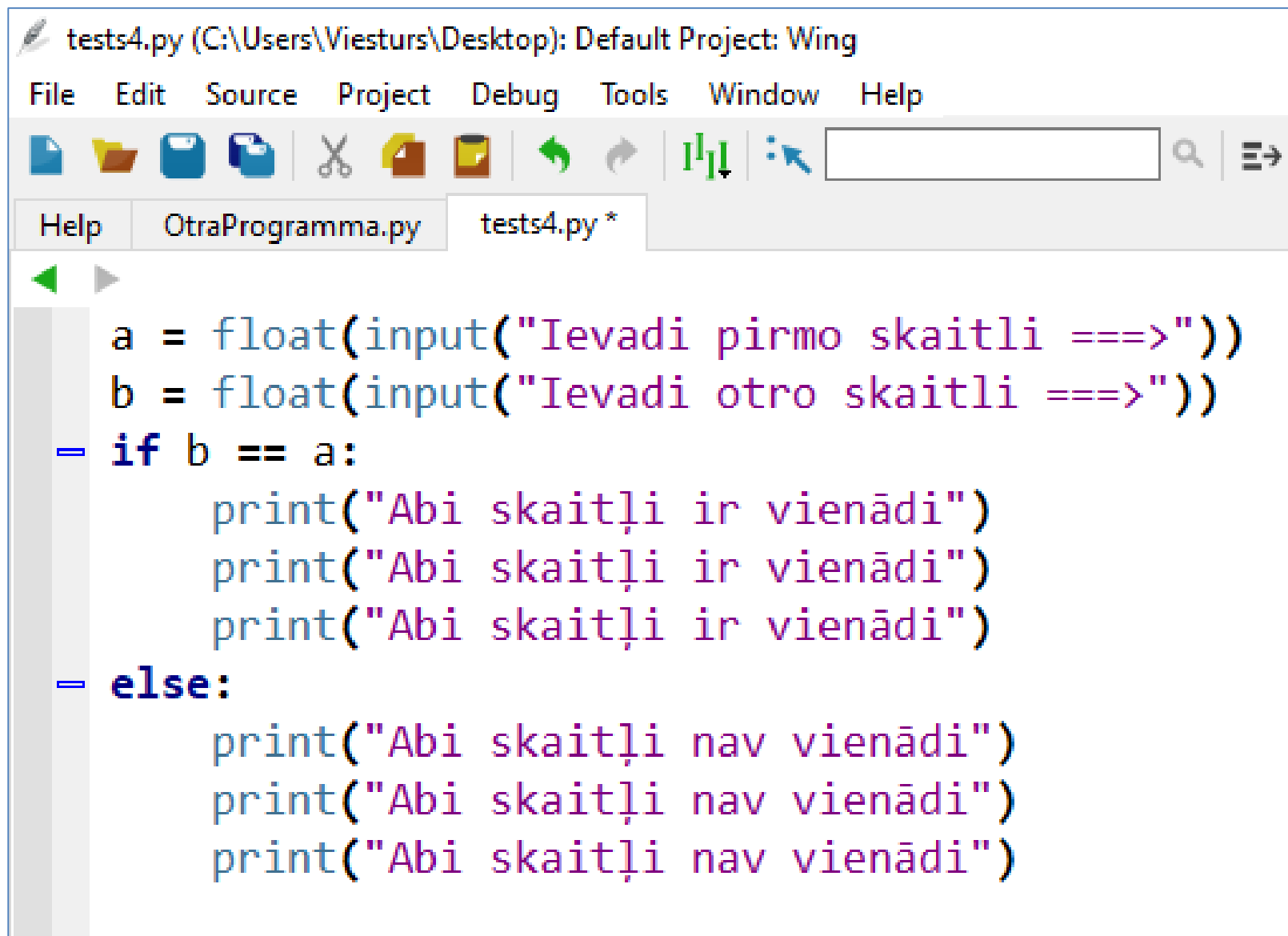
```
# Divu skaitļu salīdzināšana
```

```
a = float(input("Ievadi pirmo skaitli ==>"))
```

```
b = float(input("Ievadi otro skaitli ==>"))
```

```
print("ir") if b == a else print("nav")
```

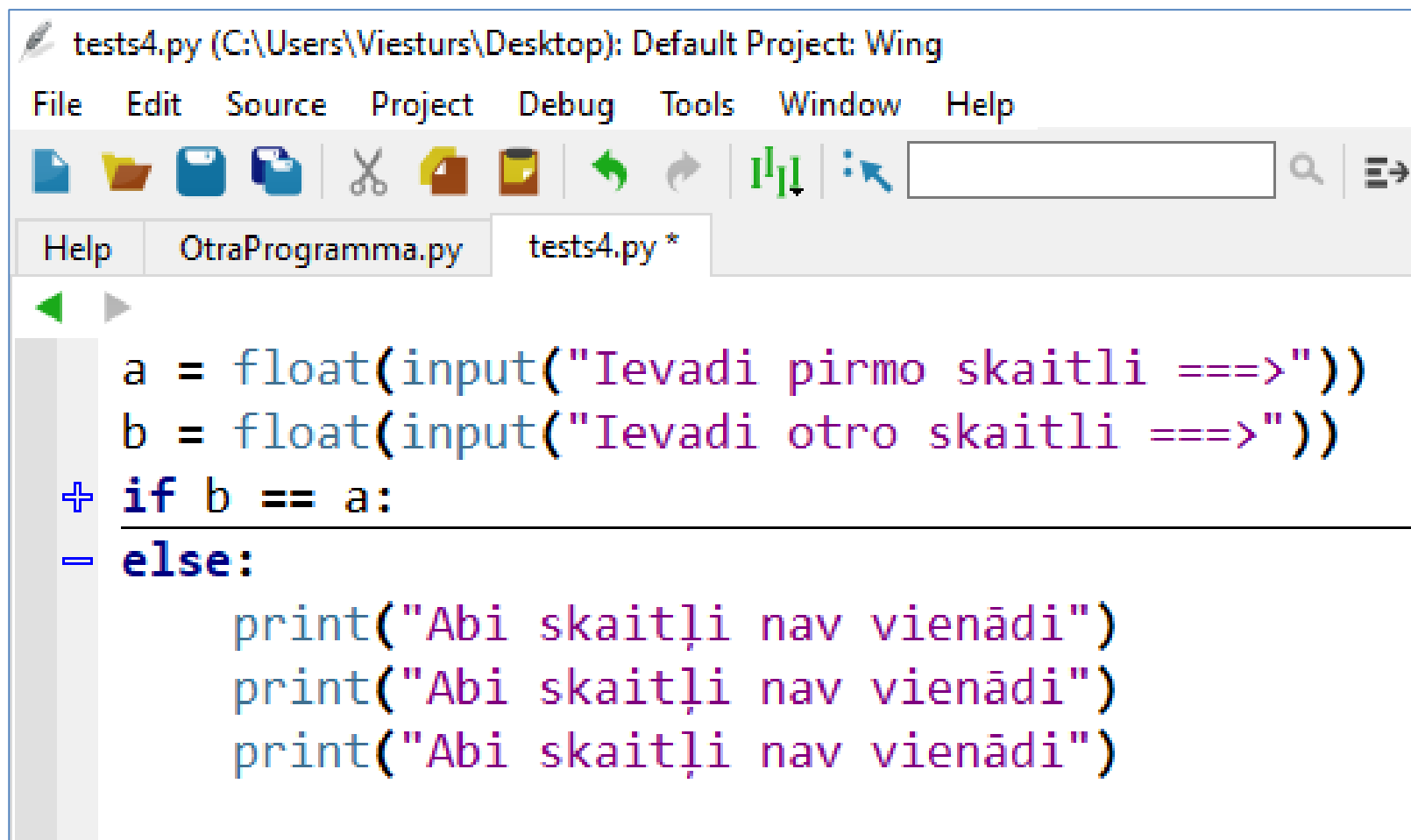
Atkāpes un "bloki"



The screenshot shows the Wing IDE interface. The title bar indicates the file is 'tests4.py' located at 'C:\Users\Viesturs\Desktop', with the project set to 'Default Project' and the editor 'Wing'. The menu bar includes 'File', 'Edit', 'Source', 'Project', 'Debug', 'Tools', 'Window', and 'Help'. The toolbar contains various icons for file operations and development. The tab bar shows 'Help', 'OtraProgramma.py', and 'tests4.py *'. The code editor displays the following Python code:

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
= if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
= else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

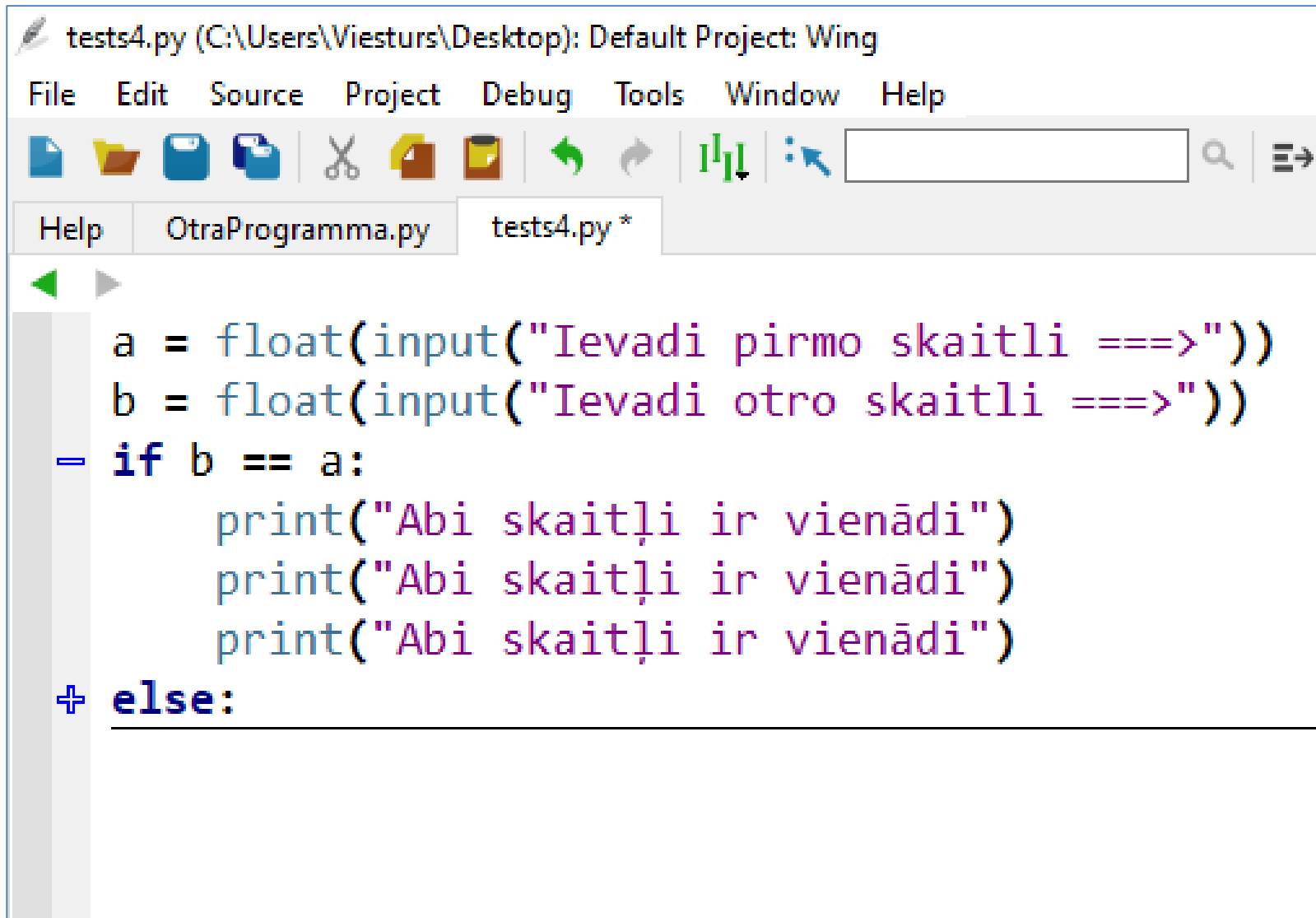
Atkāpes un "bloki"



The screenshot shows the Wing IDE interface. The title bar indicates the file is 'tests4.py' located at 'C:\Users\Viesturs\Desktop' and the project is 'Default Project'. The menu bar includes 'File', 'Edit', 'Source', 'Project', 'Debug', 'Tools', 'Window', and 'Help'. The toolbar contains various icons for file operations and development. The tab bar shows 'Help', 'OtraProgramma.py', and 'tests4.py *'. The code editor displays the following Python code:

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
+ if b == a:
- else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

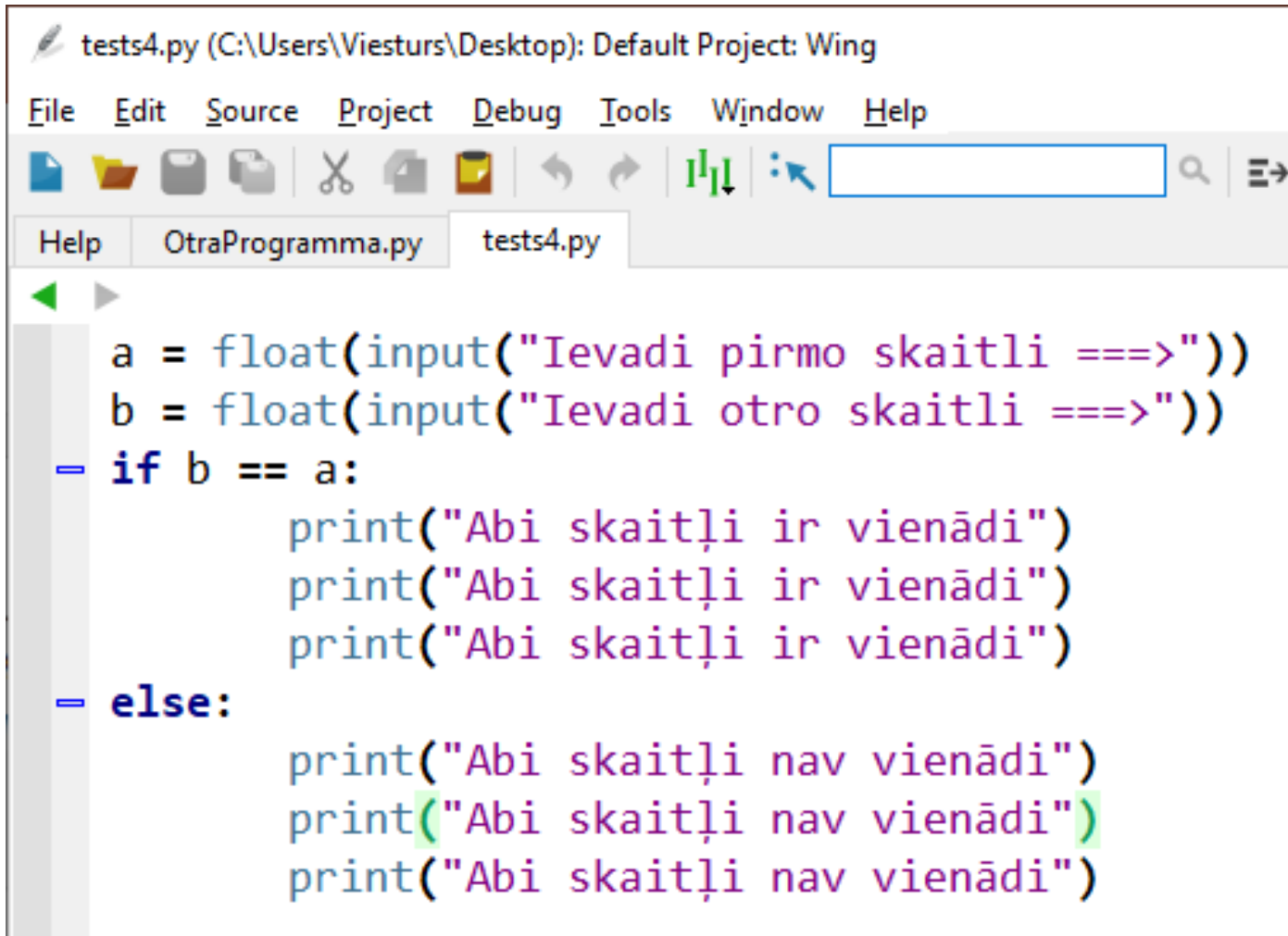
Atkāpes un "bloki"



The screenshot shows the Wing IDE interface. The title bar indicates the file is 'tests4.py' located at 'C:\Users\Viesturs\Desktop', with 'Default Project: Wing'. The menu bar includes 'File', 'Edit', 'Source', 'Project', 'Debug', 'Tools', 'Window', and 'Help'. The toolbar contains icons for file operations (new, open, save, close, copy, paste, undo, redo), a search icon, and a run icon. The tab bar shows 'Help', 'OtraProgramma.py', and 'tests4.py *'. The code editor displays the following Python code:

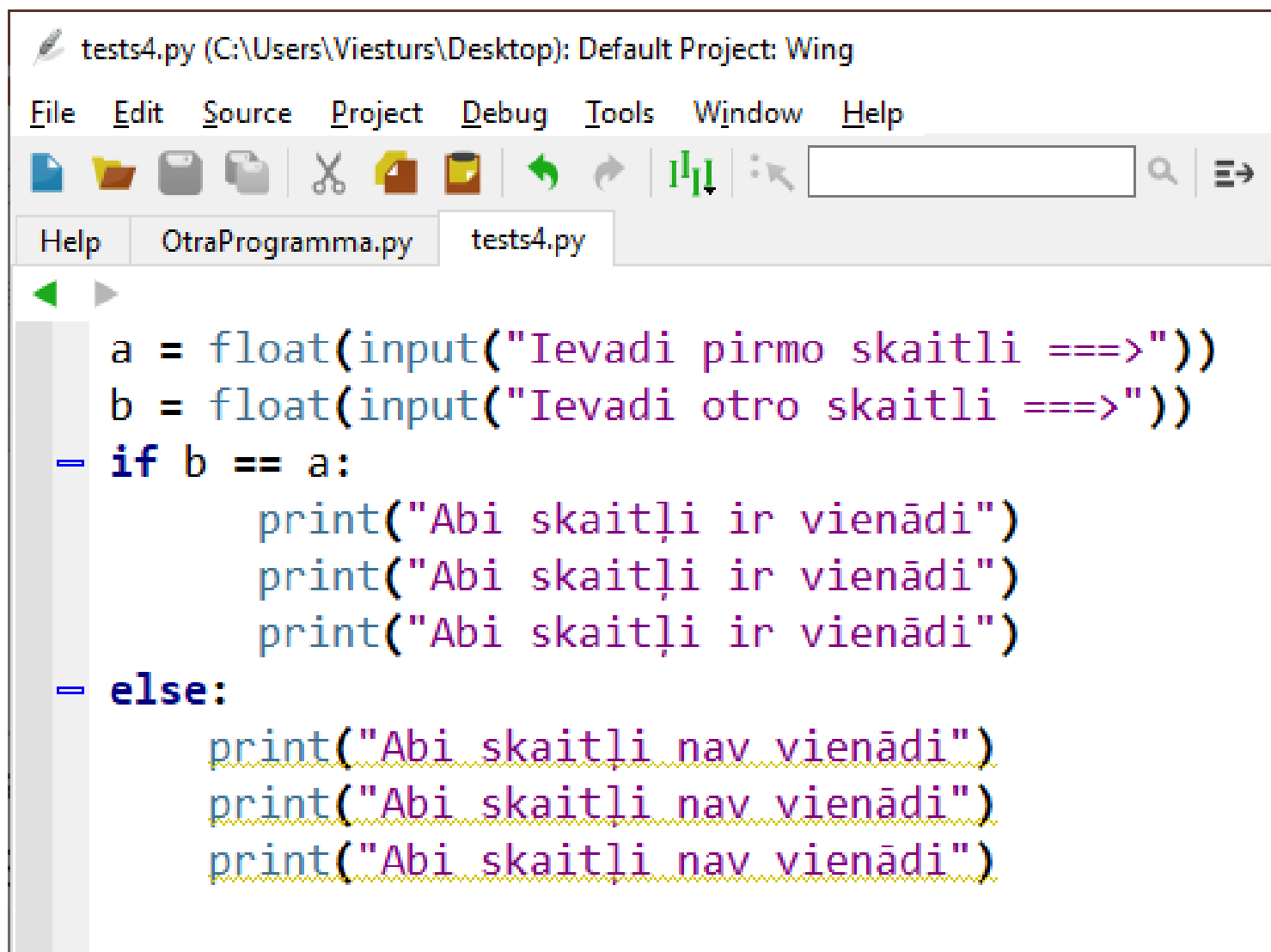
```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
= if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
+ else:
```

Atkāpes un "bloki"



```
tests4.py (C:\Users\Viesturs\Desktop): Default Project: Wing
File Edit Source Project Debug Tools Window Help
Help OtrProgramma.py tests4.py
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
= if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
= else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

Atkāpes un "bloki"

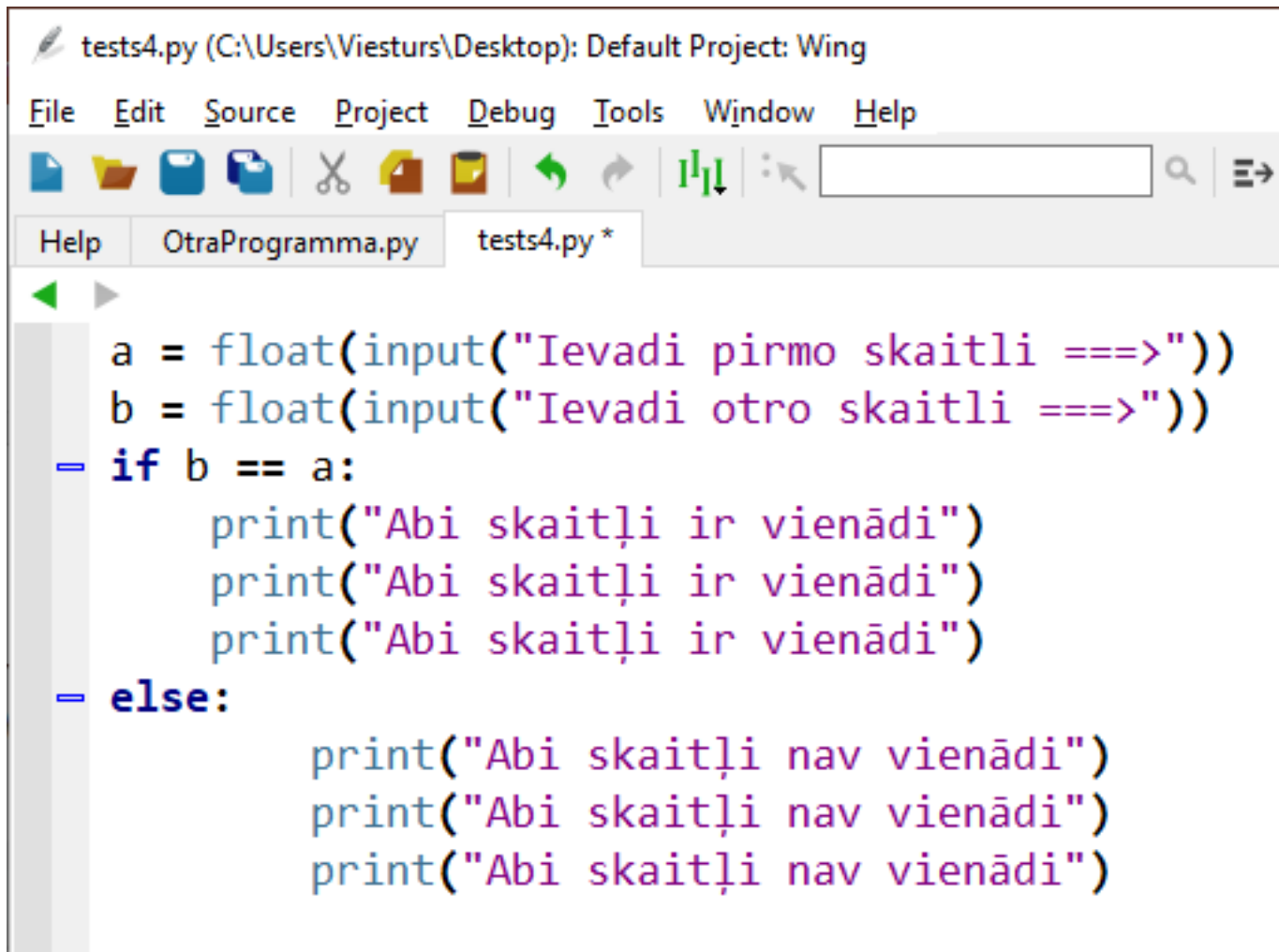


The screenshot shows the Wing IDE interface. The title bar indicates the file is 'tests4.py' located at 'C:\Users\Viesturs\Desktop' and it's the 'Default Project' in 'Wing'. The menu bar includes 'File', 'Edit', 'Source', 'Project', 'Debug', 'Tools', 'Window', and 'Help'. The toolbar contains icons for file operations (new, open, save, close, copy, paste, undo, redo), a search icon, and a run icon. The tab bar shows 'Help', 'OtraProgramma.py', and 'tests4.py'. The code editor displays the following Python code:

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
= if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
= else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

Tā nevajadzētu darīt ...

Atkāpes un "bloki"



The screenshot shows the Wing IDE interface. The title bar reads "tests4.py (C:\Users\Viesturs\Desktop): Default Project: Wing". The menu bar includes File, Edit, Source, Project, Debug, Tools, Window, and Help. The toolbar contains icons for file operations and execution. The tab bar shows "Help", "OtraProgramma.py", and "tests4.py *". The code editor displays the following Python code:

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
= if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
= else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

Arī tā nevajadzētu darīt ...

Atkāpes un "bloki"

tests4.py (C:\Users\Viesturs\Desktop): Default Project: Wing

File Edit Source Project Debug Tools Window Help

Help OtrProgramma.py tests4.py

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

Search in Files Search Stack Data tests4.py, line 5 Options

Variable	Value
globals	<globals dict; len=0>

Debug I/O Exceptions Python Shell Messages

☐ Ignore this exception location Options

File "C:\Users\Viesturs\Desktop\tests4.py", line 5, in ?
print("Abi skaitļi ir vienādi")

builtins.IndentationError: unexpected indent (tests4.py, line 5)

Line 5 Col 0 - [User]

Atkāpes un "bloki"

The screenshot shows the Wing IDE interface with a Python script named `tests4.py`. The script contains the following code:

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

The `if b == a:` line is highlighted in red. The IDE's status bar at the bottom left shows "Line 3 Col 0 - [User]".

The bottom panel displays the "Messages" tab, showing an error message:

File "C:\Users\Viesturs\Desktop\tests4.py", line 3, in ?
`if b == a:`
builtins.IndentationError: unexpected indent (tests4.py, line 3)

Atkāpes un "bloki"

The screenshot shows the Wing IDE interface with a Python script named `tests4.py` open. The script contains the following code:

```
a = float(input("Ievadi pirmo skaitli ==>"))
b = float(input("Ievadi otro skaitli ==>"))
if b == a:
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
    print("Abi skaitļi ir vienādi")
else:
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
    print("Abi skaitļi nav vienādi")
```

The first line of the script is highlighted in red. The IDE's status bar at the bottom left shows "Line 1 Col 0 - [User]".

The bottom right pane displays the "Messages" tab, showing an exception:

builtins.IndentationError: unexpected indent (tests4.py, line 1)

The message text indicates the error occurred in the file `"C:\Users\Viesturs\Desktop\tests4.py"`, line 1, in the function `a = float(input("Ievadi pirmo skaitli ==>"))`.

```
import random

# Izveidojam "nosacīto konstanti"
SK1 = int(10*random.random())
# print (SK1)

sk2=int(input("Ievadi skaitli ==> "))

if SK1 == sk2 :
    print("Atminēts!")
else :
    print("Nav atminēts!")
```

$$f(x) = \begin{cases} x, & \text{ja } x < 0 \\ x \cdot \cos(x), & \text{ja } x \geq 0 \end{cases}$$

```
import math
```

```
x=float(input("Ievadi skaitli ==> "))
```

```
if x < 0 :
```

```
    y = x
```

```
else :
```

```
    y = x * math.cos(x)
```

```
print("F(",str(x), ")=" , str(y))
```

$$f(x) = \begin{cases} x^3, & \text{ja } x < 1 \\ x^2, & \text{ja } x \geq 1 \end{cases}$$

```
x=float(input("Ievadi skaitli ==> "))
```

```
if x < 1 :
```

```
    y = x ** 3    # y=x*x*x
```

```
else :
```

```
    y = x * x
```

```
print("F(",str(x), ")=" , str(y))
```

NB! Paskaidrot par int un float šajā gadījumā

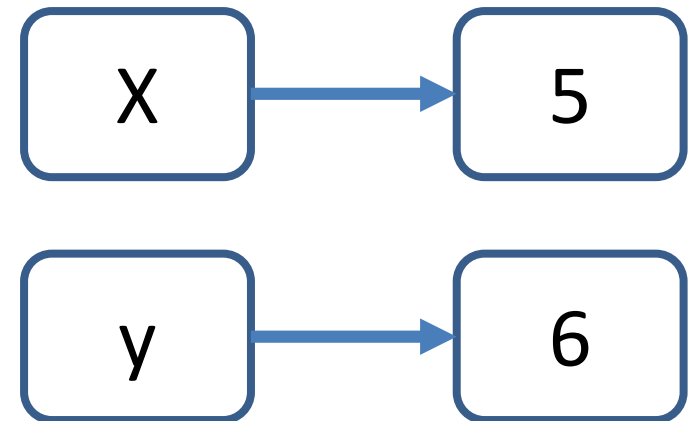
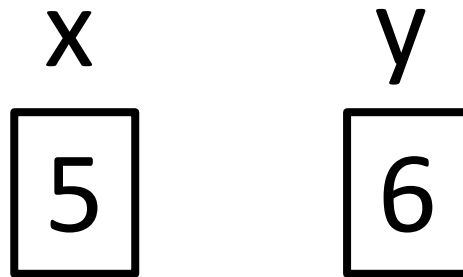
Trīs ārēji līdzīgu akmeņu
salīdzināšana pēc masas

```
a = int(input("Ievadiet 1.skaitli ==> "))  
b = int(input("Ievadiet 2.skaitli ==> "))  
c = int(input("Ievadiet 3.skaitli ==> "))
```

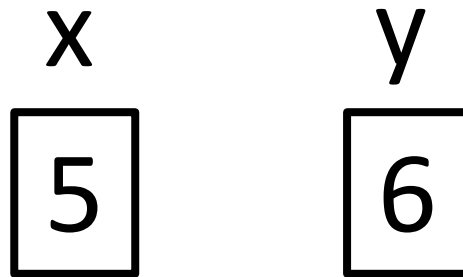
```
if a < b :  
    x = a  
else :  
    x = b  
if x > c :  
    x = c
```

```
print("Mazākais skaitlis ir:", str(x))
```


Divu mainīgo vērtību samainīšana vietām



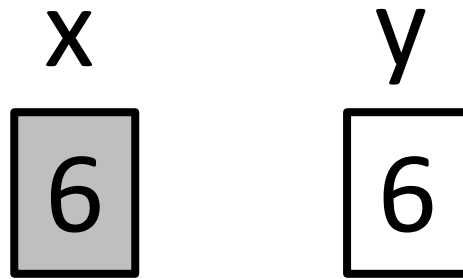
Divu mainīgo vērtību samainīšana vietām



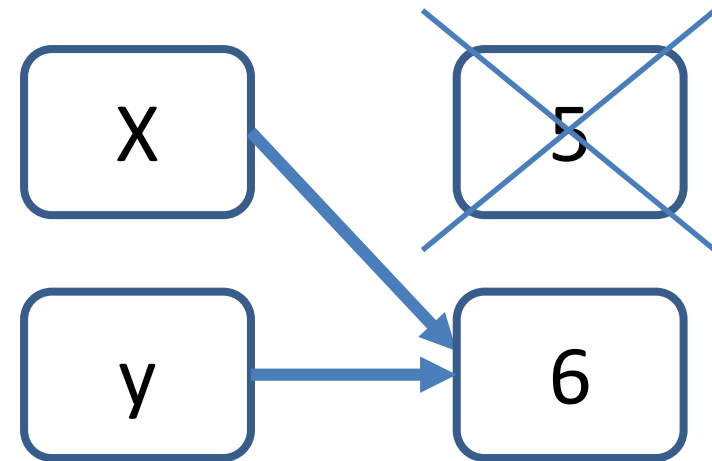
$$x = y$$

$$y = x$$

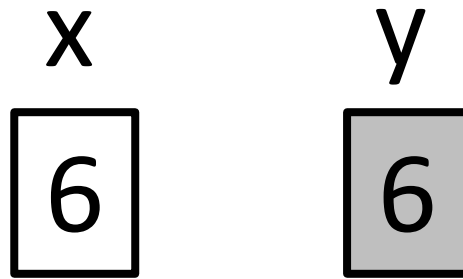
Divu mainīgo vērtību samainīšana vietām



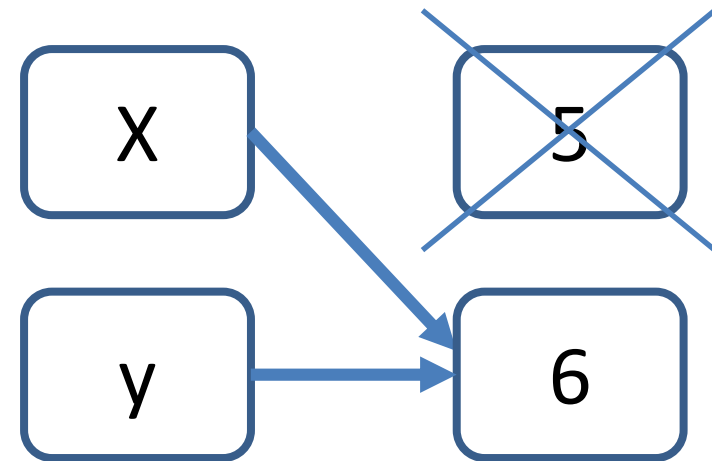
$$x = y$$



Divu mainīgo vērtību samainīšana vietām



$x = y$
 $y = x$

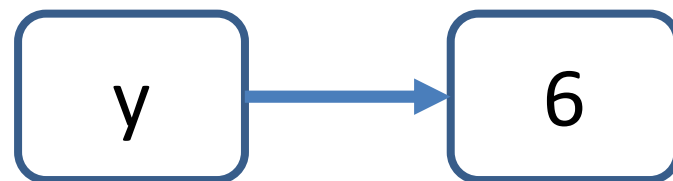
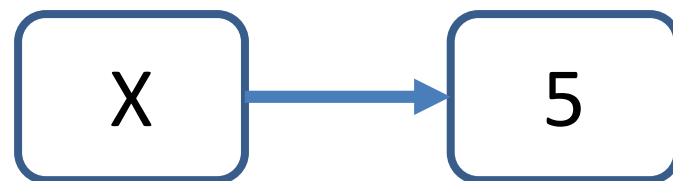


Divu mainīgo vērtību samainīšana vietām

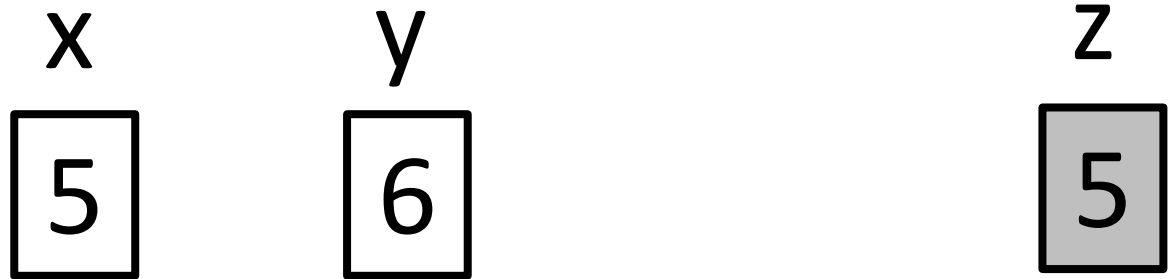
x
5

y
6

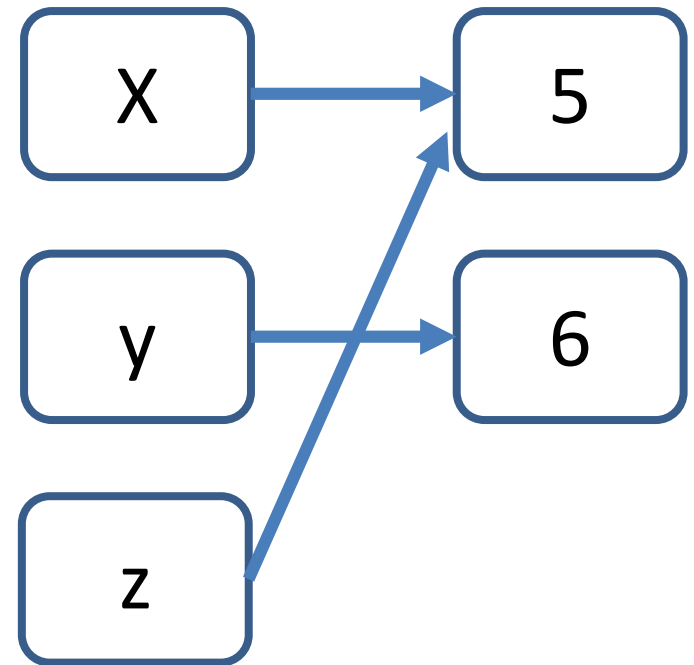
~~z
0~~



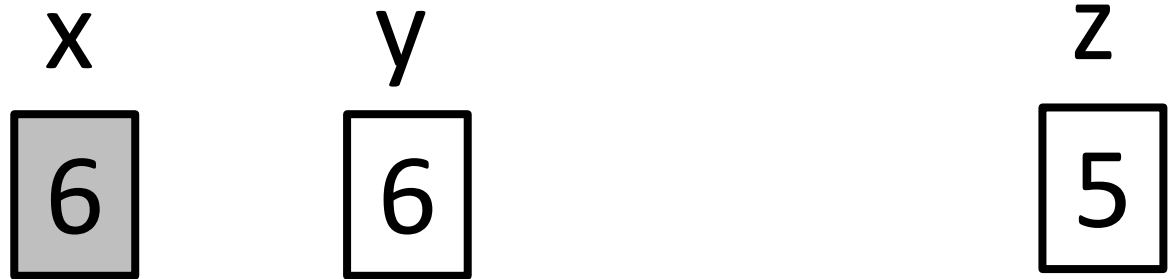
Divu mainīgo vērtību samainīšana vietām



$z = x$

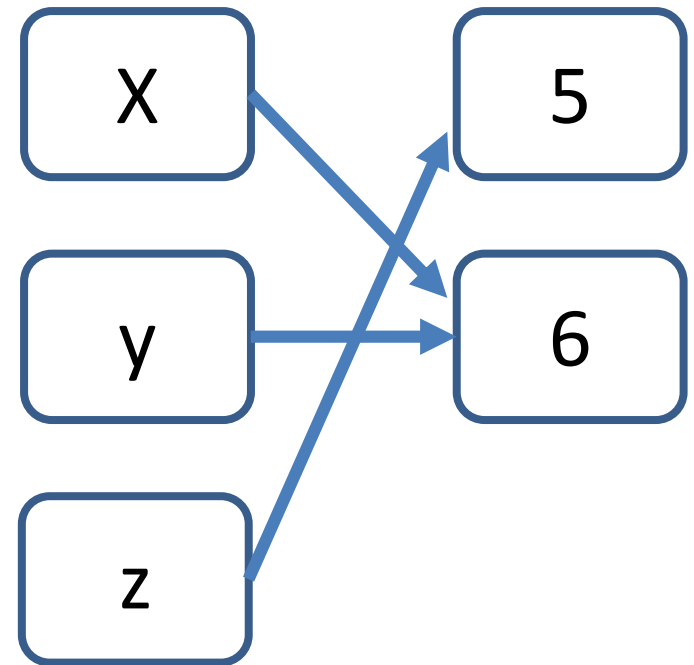


Divu mainīgo vērtību samainīšana vietām



$z = x$

$x = y$



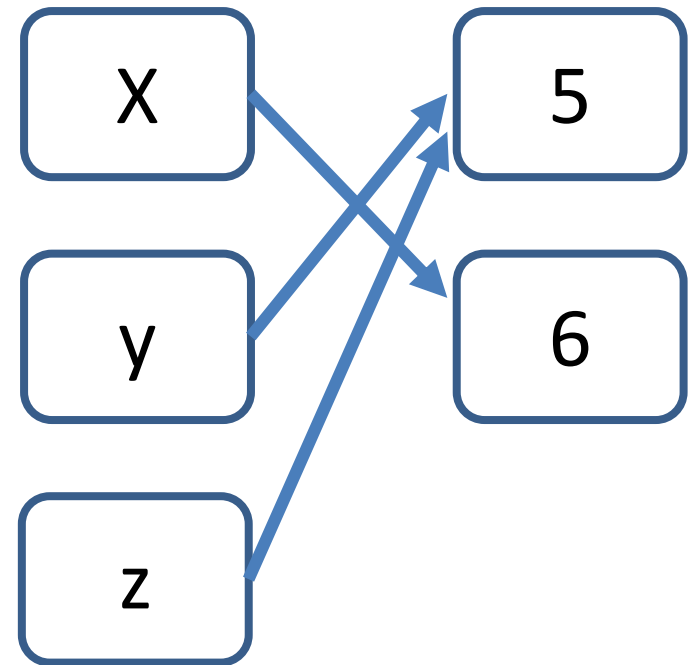
Divu mainīgo vērtību samainīšana vietām



$z = x$

$x = y$

$y = z$




```
a = int(input("Ievadiet 1.skaitli ==> "))  
b = int(input("Ievadiet 2.skaitli ==> "))  
c = int(input("Ievadiet 3.skaitli ==> "))
```

```
if a > b :
```

```
    x = a
```

```
    a = b
```

```
    b = x
```

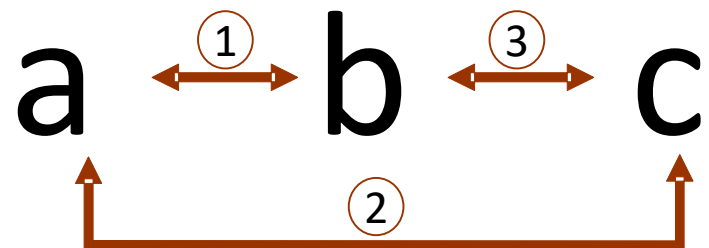
```
if a > c :
```

```
    x = c
```

```
    c = a
```

```
    a = x
```

```
print(str(a), str(b),str(c))
```



```
a = int(input("Ievadiet 1.skaitli ==> "))  
b = int(input("Ievadiet 2.skaitli ==> "))  
c = int(input("Ievadiet 3.skaitli ==> "))
```

```
if a > b :
```

```
    x = a
```

```
    a = b
```

```
    b = x
```

```
if a > c :
```

```
    x = c
```

```
    c = a
```

```
    a = x
```

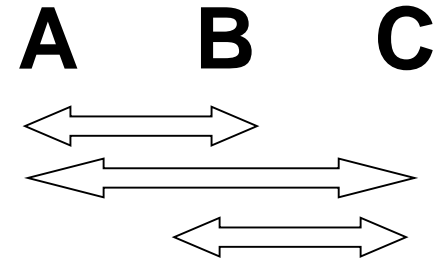
```
if b > c :
```

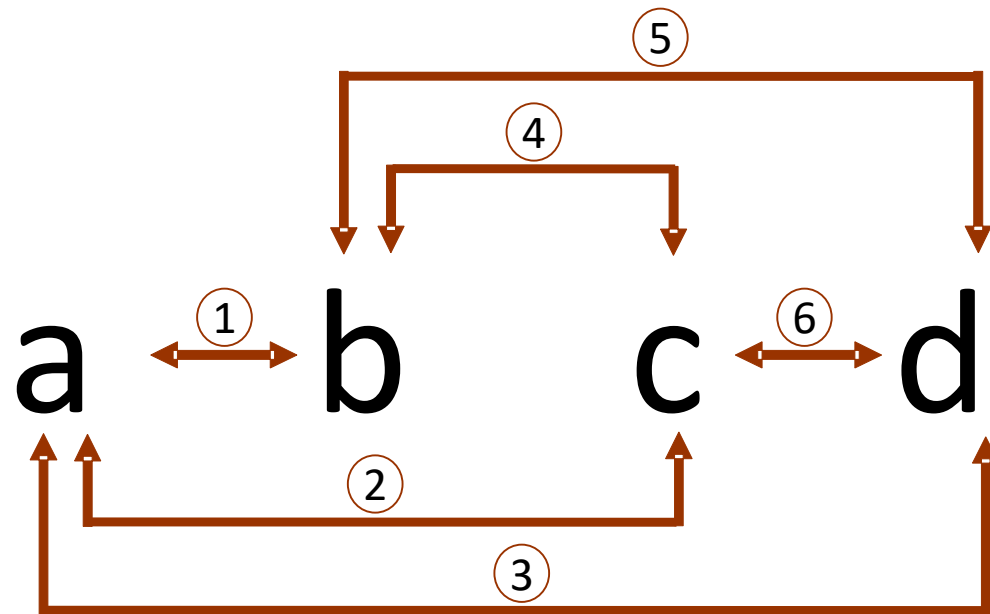
```
    x = b
```

```
    b = c
```

```
    c = x
```

```
print(str(a), str(b),str(c))
```





```
a = int(input("Ievadiet 1.skaitli ==> "))
b = int(input("Ievadiet 2.skaitli ==> "))
c = int(input("Ievadiet 3.skaitli ==> "))
d = int(input("Ievadiet 4.skaitli ==> "))
```

```
if a > b :
```

```
    x = a
```

```
    a = b
```

```
    b = x
```

```
if a > c :
```

```
    x = c
```

```
    c = a
```

```
    a = x
```

```
if a > d :
```

```
    x = d
```

```
    d = a
```

```
    a = x
```

```
if b > c :
```

```
    x = b
```

```
    b = c
```

```
    c = x
```

```
if b > d :
```

```
    x = b
```

```
    b = d
```

```
    d = x
```

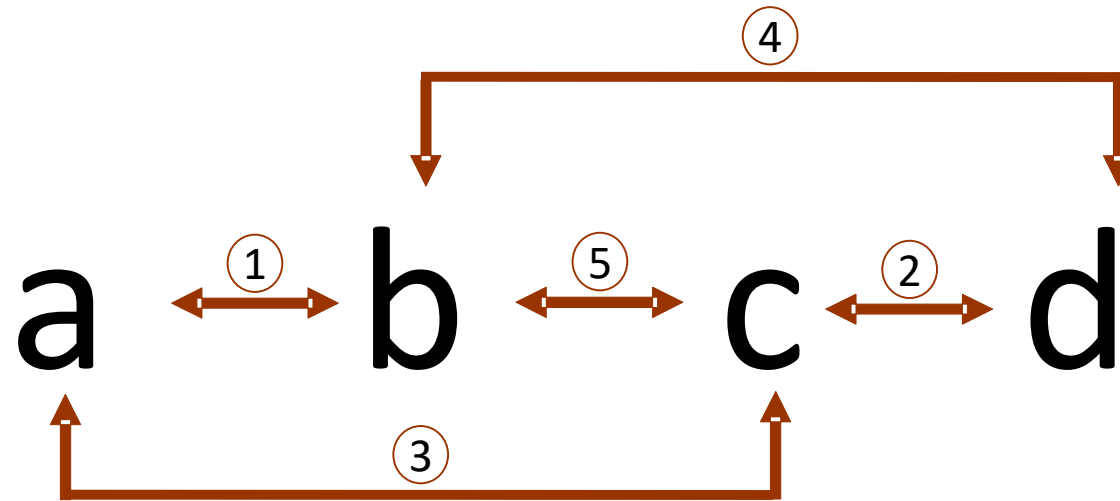
```
if c > d :
```

```
    x = c
```

```
    c = d
```

```
    d = x
```

```
print(str(a), str(b), str(c), str(d))
```



```
a = int(input("Ievadiet 1.skaitli ==> "))  
b = int(input("Ievadiet 2.skaitli ==> "))  
c = int(input("Ievadiet 3.skaitli ==> "))  
d = int(input("Ievadiet 4.skaitli ==> "))
```

```
if a > b :
```

```
    x = a
```

```
    a = b
```

```
    b = x
```

```
if c > d :
```

```
    x = c
```

```
    c = d
```

```
    d = x
```



```
if a > c :
```

```
    x = c
```

```
    c = a
```

```
    a = x
```

```
if b > d :
```

```
    x = b
```

```
    b = d
```

```
    d = x
```

```
if b > c :
```

```
    x = b
```

```
    b = c
```

```
    c = x
```

```
print(str(a), str(b), str(c), str(d))
```

Izcilības (desmitnieka) uzdevums

- Noskaidrot, kāds ir mazākais salīdzināšanu skaits, lai 8 skaitļus sakārtotu augošā secībā. Uzzīmēt blokshēmu un uzrakstīt programmu, kas realizē 8 skaitļus sakārtošanu augošā secībā ar mazāko iespējamo salīdzināšanu skaitu. Atļauts izmantot tikai ievades, izvades un zarošanās priekšrakstus, salīdzināšanas un piešķires operatorus (7%)
- Pierādīt, ka ar mazāku salīdzināšanu skaitu nav iespējams veikt 8 skaitļu sakārtošanu augošā secībā. (7%)

Vai var izveidot trijstūri

```
a = int(input("Ievadiet 1.skaitli ==> "))
b = int(input("Ievadiet 2.skaitli ==> "))
c = int(input("Ievadiet 3.skaitli ==> "))
```

```
if a < b :
```

```
    x = a
```

```
    a = b
```

```
    b = x
```

```
if a < c :
```

```
    x = c
```

```
    c = a
```

```
    a = x
```

```
if (a < b + c) :
```

```
    print("Var izveidot trijsturi")
```

```
else :
```

```
    print("Nevar izveidot trijsturi")
```

```
a = int(input("Ievadiet 1.skaitli ==> "))
b = int(input("Ievadiet 2.skaitli ==> "))
c = int(input("Ievadiet 3.skaitli ==> "))

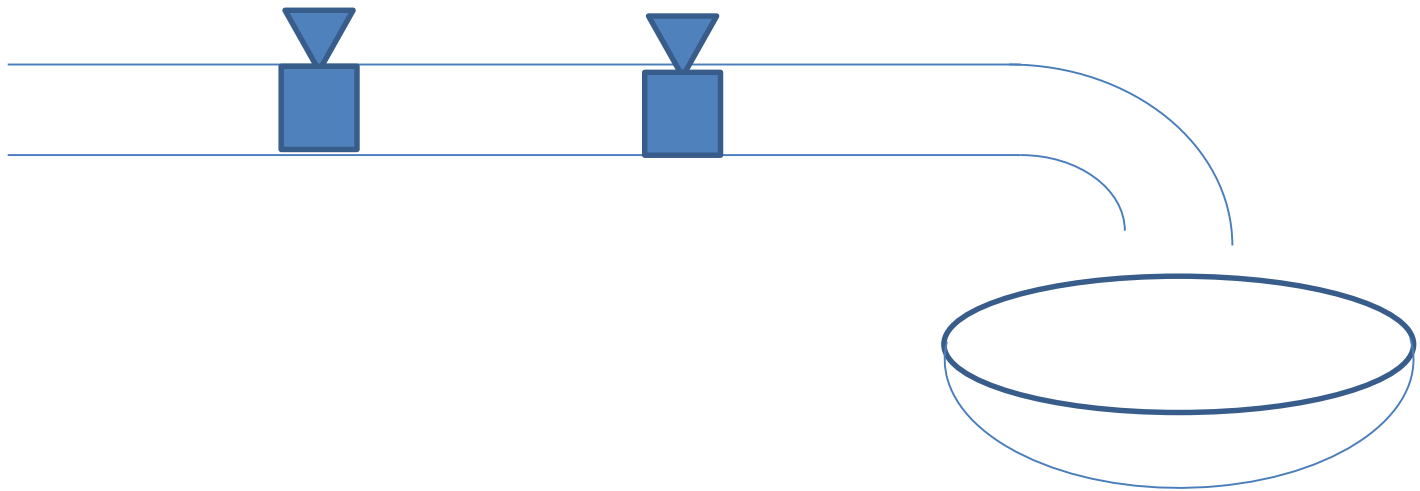
if (a + b > c) and (b + c > a) and (a + c > b) :
    print("Var izveidot trijsturi")
else :
    print("Nevar izveidot trijsturi")
```

Loģiskās operācijas

- not – negācija
- and – konjunkcija
- or – disjunktija

X	Y	NOT X	X AND Y	X OR Y
FALSE	FALSE	TRUE	FALSE	FALSE
FALSE	TRUE	TRUE	FALSE	TRUE
TRUE	FALSE	FALSE	FALSE	TRUE
TRUE	TRUE	FALSE	TRUE	TRUE

X and Y



$$a = b = c$$

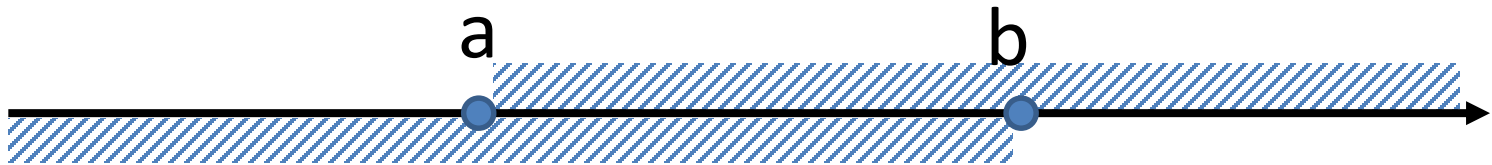
$$a = b, a = c, b = c$$

$$a = b \text{ and } b = c$$
$$(a = b) \text{ and } (b = c)$$

$$x \in [a, b]$$

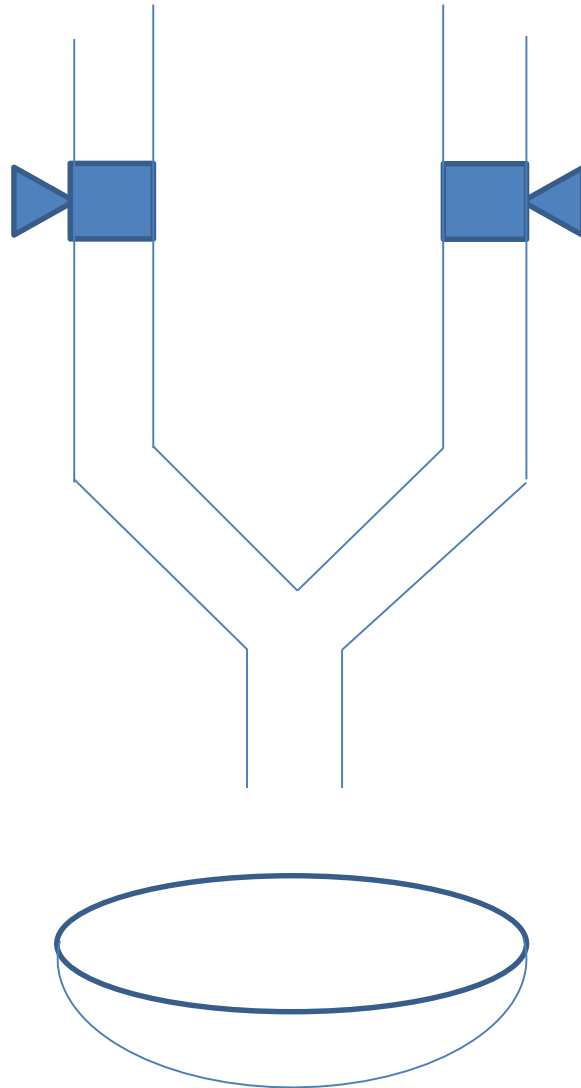
jeb

$$a \leq x \leq b$$



$$(a \leq x) \text{ and } (x \leq b)$$

X or Y



$$x \notin [a, b]$$

jeb

$$x \in (-\infty, a) \cup (b, +\infty)$$

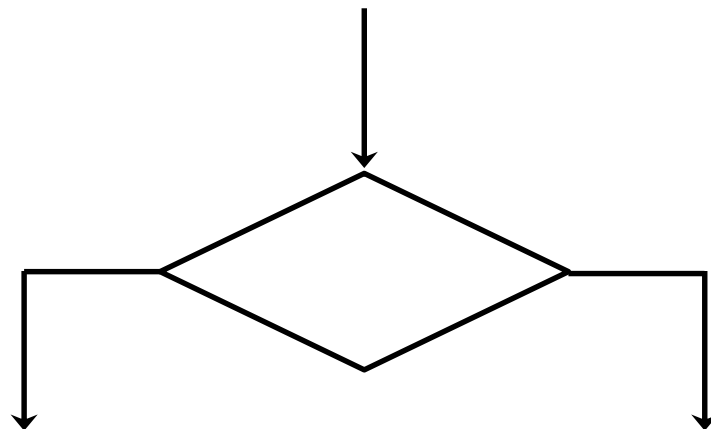
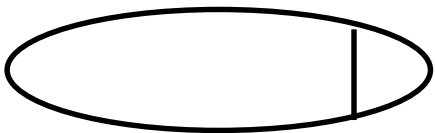
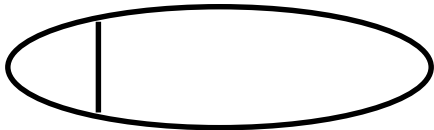


$$\text{not } ((a \leq x) \text{ and } (x \leq b))$$

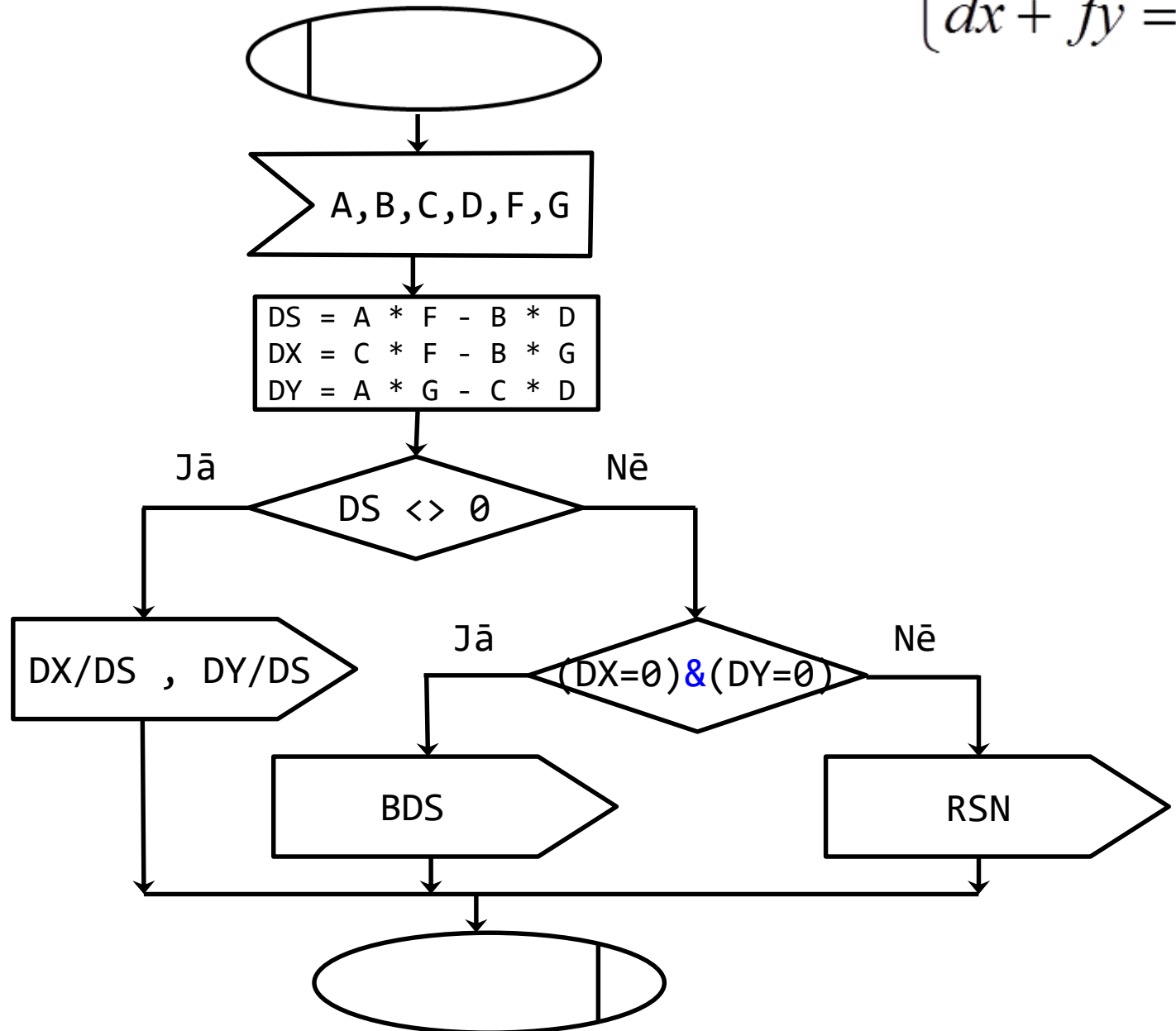
jeb

$$(x < a) \text{ or } (x > b)$$

Blokshēma



$$\begin{cases} ax + by = c \\ dx + fy = g \end{cases}$$



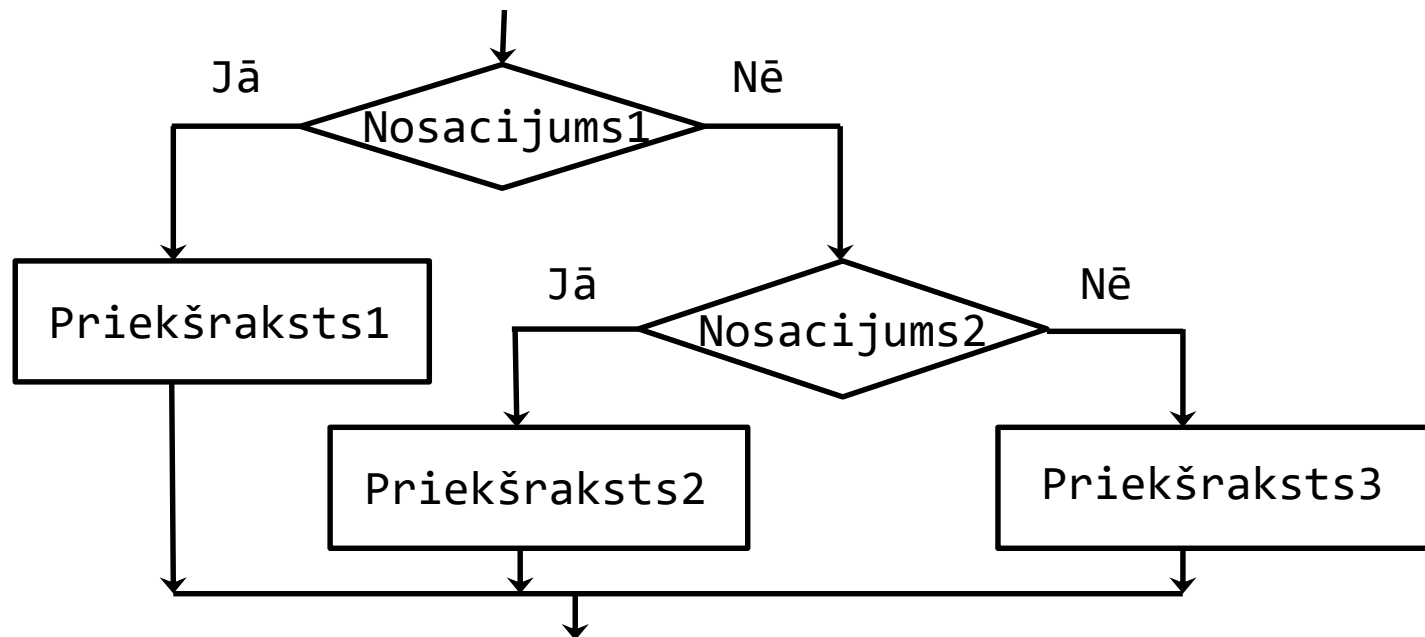
$$\begin{cases} ax + by = c \\ dx + fy = g \end{cases}$$

```
a = int(input("Ievadiet koeficientu a ==> "))
b = int(input("Ievadiet koeficientu b ==> "))
c = int(input("Ievadiet koeficientu c ==> "))
d = int(input("Ievadiet koeficientu d ==> "))
f = int(input("Ievadiet koeficientu f ==> "))
g = int(input("Ievadiet koeficientu g ==> "))
```

```
ds = a * f - b * d
dx = c * f - b * g
dy = a * g - c * d
```

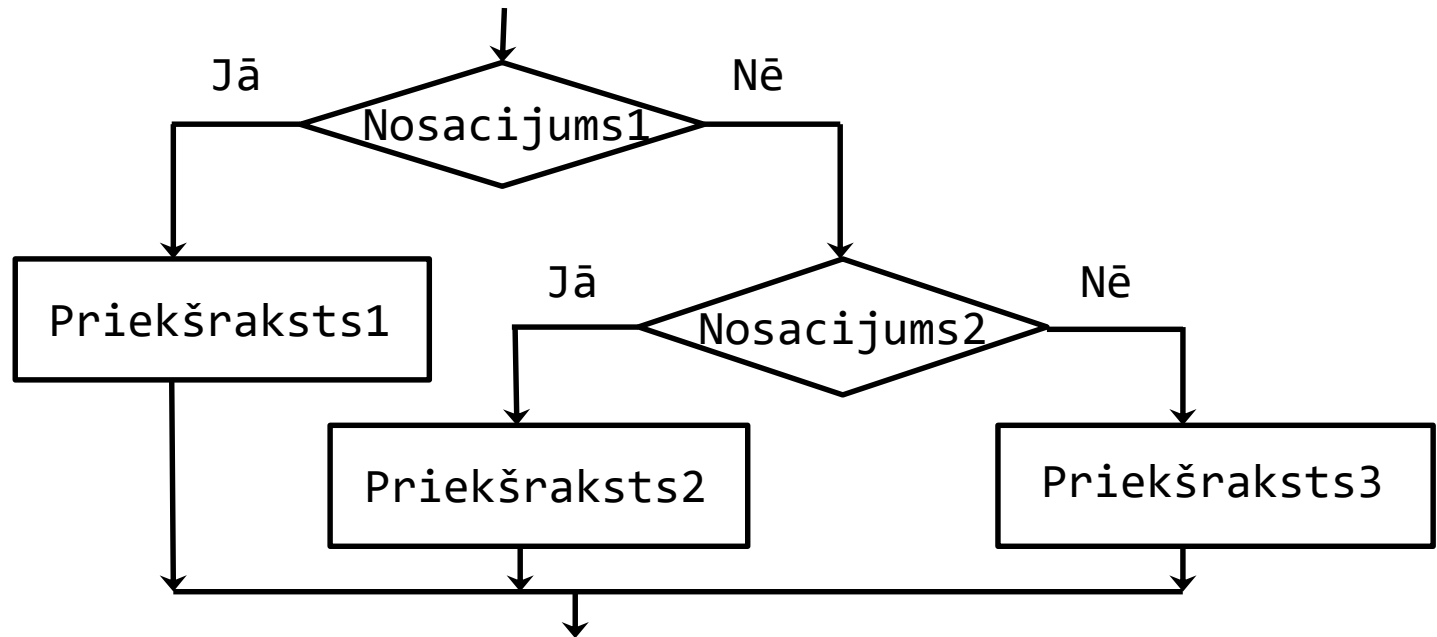
```
if ds != 0 :  
    x = dx / ds  
    y = dy / ds  
    print("VVA. (" + str(x) + " , " + str(y) + ") ")  
  
else :  
    if (dx == 0) and (dy == 0) :  
        print("BEZGALIGI DAUDZ ATRISINAJUMU")  
    else :  
        print("ATRISINAJUMU NAV")
```

```
if Nosacījums1 :  
    Priekšraksts1  
else  
    if Nosacījums2 :  
        Priekšraksts2  
    else :  
        Priekšraksts3
```

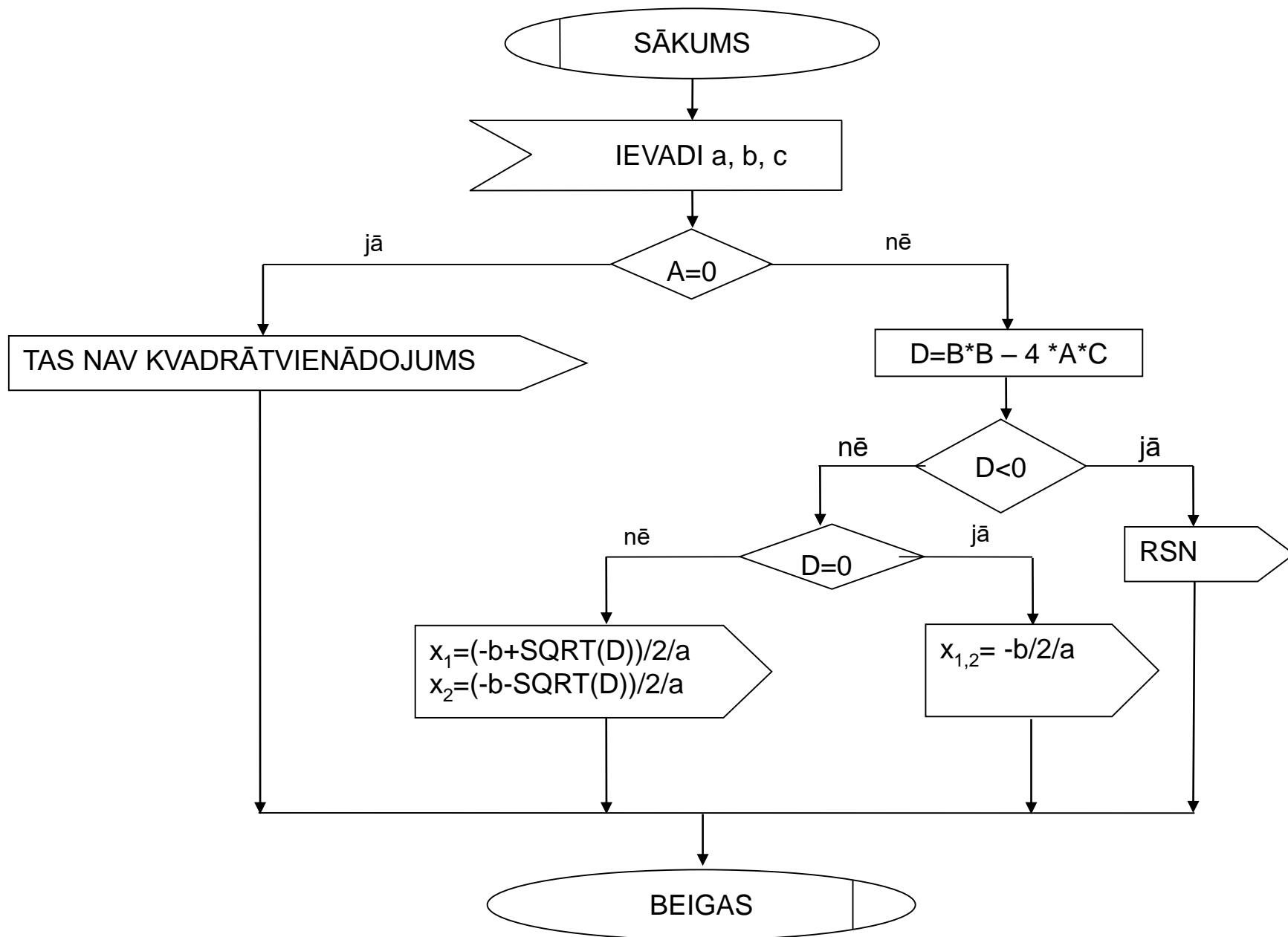



```
if ds != 0 :  
    x = dx / ds  
    y = dy / ds  
    print("VVA. (" + str(x) + " , " + str(y) + ") ")  
  
elif (dx == 0) and (dy == 0) :  
    print("BEZGALIGI DAUDZ ATRISINAJUMU")  
  
else :  
    print("ATRISINAJUMU NAV")
```

```
if Nosacījums1 :  
    Priekšraksts1  
elif Nosacījums2 :  
    Priekšraksts2  
else :  
    Priekšraksts3
```



```
if nosacījums1 :  
    priekšraksts1  
elif nosacījums2 :  
    priekšraksts2  
  
    . . .  
  
else :  
    priekšrakstsN
```



```
import math
a = int(input("Ievadiet koeficientu a ==> "))
b = int(input("Ievadiet koeficientu b ==> "))
c = int(input("Ievadiet koeficientu c ==> "))
if a == 0 :
    print("Tas nav kvadratvienadojums")
else :
    d = b * b - 4 * a * c # Diskriminants
    if d < 0 :
        print("Kvadratvienādojumam realu saknu nav")
    elif d == 0 :
        x12 = -b / 2 / a
        print("X1=X2=" + str(x12))
    else :
        x1 = (-b + math.sqrt(d)) / (2 * a)
        x2 = (-b - math.sqrt(d)) / 2 / a
        print("X1=" + str(x1))
        print("X2=" + str(x2))
```

Programmatūras testēšana

- Programmatūras testēšana ir tās darbības pārbaude
- Testēšanas mērķis var būt:
 - defekta atklāšana
 - defekta atrašanās vietas lokalizēšana
 - testējamā objekta dinamisko parametru (piemēram, ātrdarbības) noskaidrošana
- Tests ir viena vai vairāku testpiemēru kopa apvienojumā ar testpiemēru izpildes procedūru, kas domāta kādas programmas īpašības pārbaudei
- Absolūta testēšana nav iespējama

Testēšanas metodes

- Melnās kastes testēšana, kura balstās uz programmas pārbaudi, zinot kādas funkcijas tajā ir paredzētas, bet nezinot to iekšējo struktūru
- Baltās kastes testēšana, kura balstās uz programmas pārbaudi, zinot gan programmas funkcijas, gan iekšējo struktūru
- Pelēkās kastes testēšana, kura balstās uz programmas pārbaudi, apvienojot melnās un baltās kastes testēšanas principus

Melnās kastes testēšana

- Melnās kastes testēšana vai uzvedības testēšana, fokusējas uz programmatūras funkcionālajām prasībām
- Tiek izvēlēti tādi testpiemēri, kas iespējami pilnīgi pārbauda funkcionālās prasības
- Melnās kastes testēšana paredzēta šādu kļūdu kategoriju atklāšanai:
 - nepareizas vai trūkstošas funkcijas
 - saskarnes kļūdas
 - datu struktūru kļūdas vai ārējas datubāzes piekļuves kļūdas
 - uzvedības vai veiktspējas (performance) kļūdas
 - inicializācijas vai nobeigšanas kļūdas

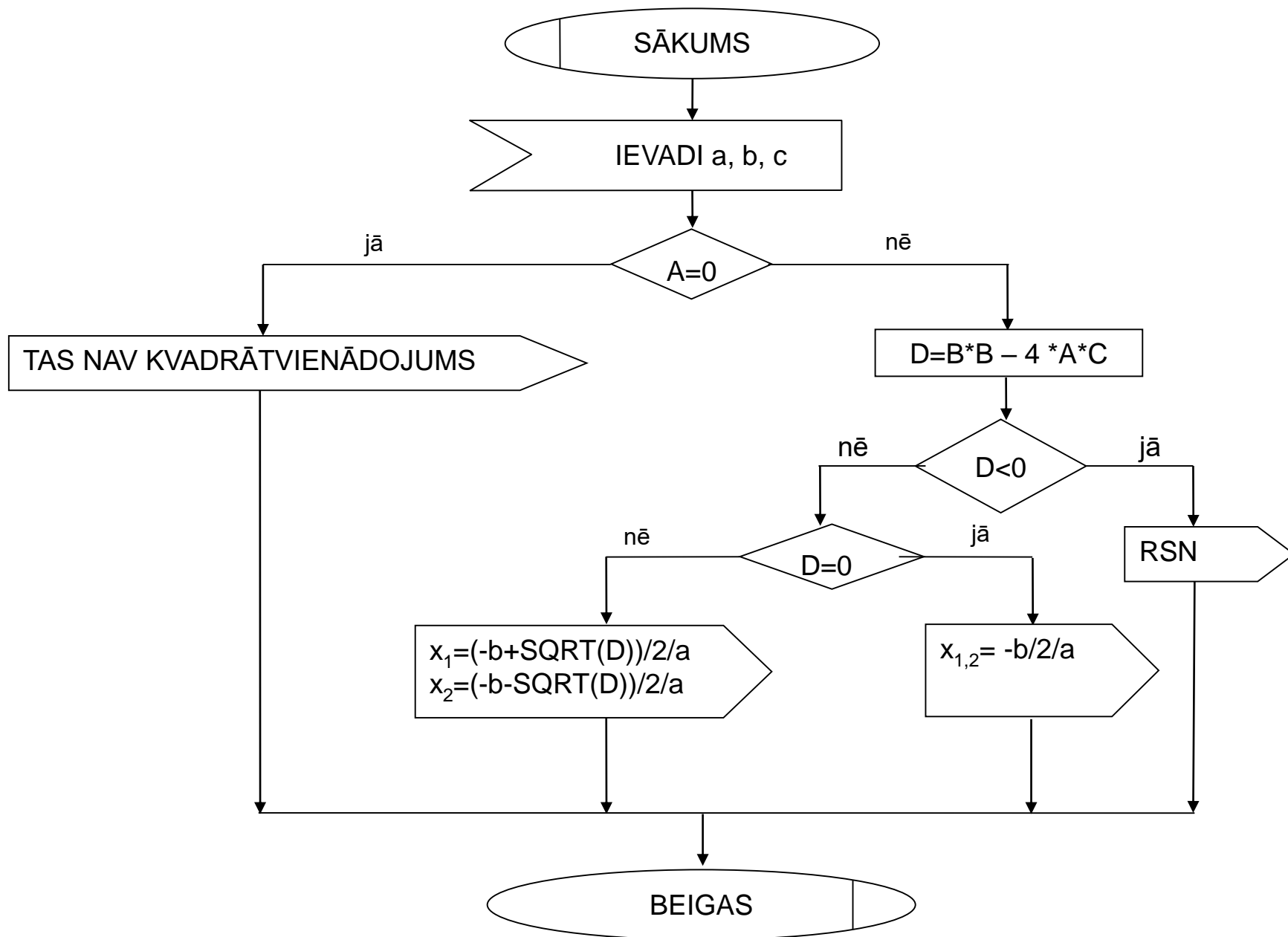
Baltās kastes testēšana

- Baltās kastes testēšana balstās uz testu projektēšanas principu, kas izmanto programmas vadības struktūras elementu pārklāšanu ar testiem.
- Testu izvēles principi:
 - katrs neatkarīgs programmas ceļš tiktu iziets vismaz vienu reizi
 - vismaz pa reizei tiktu iziets katrs programmas sazarojums
 - katram ciklam tiktu izieti gan tā robežgadījumi, gan tā ķermenis
 - tiktu izmantotas iekšējās datu struktūras



PROGRAMMATŪRAS TESTĒŠANAS ROKASGRĀMATA VADĪTĀJIEM

Versija 1.0



$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

```
import math
x = float(input("Ievadi skaitli ==>"))
if (x>=-2*math.pi) and (x<=2*math.pi) :
    y=x*math.sin(x)
    print("F(",str(x),")=",str(y))
if x<-7 :
    y=abs(x)
    print("F(",str(x),")=",str(y))
if x>7 :
    y=x-7
    print("F(",str(x),")=",str(y))
```

$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

```
import math
x = float(input("Ievadi skaitli ==>"))
if (x>=-2*math.pi) and (x<=2*math.pi) :
    y=x*math.sin(x)
    print("F(",str(x),")=",str(y))
elif x<-7 :
    y=abs(x)
    print("F(",str(x),")=",str(y))
else :
    y=x-7
    print("F(",str(x),")=",str(y))
```

$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

```
import math
x = float(input("Ievadi skaitli ==>"))
if (x>=-2*math.pi) and (x<=2*math.pi) :
    y=x*math.sin(x)
    print("F(",str(x),")=",str(y))
elif x<-7 :
    y=abs(x)
    print("F(",str(x),")=",str(y))
else :
    y=x-7
    print("F(",str(x),")=",str(y))
```



$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

```
import math
x = float(input("Ievadi skaitli ==>"))
if (x>=-2*math.pi) and (x<=2*math.pi) :
    y=x*math.sin(x)
    print("F(",str(x),")=",str(y))
elif x<-7 :
    y=abs(x)
    print("F(",str(x),")=",str(y))
elif x>7 :
    y=x-7
    print("F(",str(x),")=",str(y))
```



$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

```
import math
x = float(input("Ievadi skaitli ==>"))
if (x>=-2*math.pi) and (x<=2*math.pi) :
    y=x*math.sin(x)
elif x<-7 :
    y=abs(x)
elif x>7 :
    y=x-7

print("F(",str(x),")=",str(y))
```



$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

```
import math
x = float(input("Ievadi skaitli ==>"))
if (x>=-2*math.pi) and (x<=2*math.pi) :
    y=x*math.sin(x)
elif x<-7 :
    y=abs(x)
elif x>7 :
    y=x-7
print("F(",str(x),")=",str(y))
```

File "C:\Users\Viesturs\Desktop\tests4.py", line 10, in <module>

```
print("F(",str(x),")=",str(y))
```

builtins.NameError: name 'y' is not defined



$$f(x) = \begin{cases} x \cdot \sin x, & \text{ja } x \in [-2\pi; 2\pi] \\ |x|, & \text{ja } x < -7 \\ x - 7, & \text{ja } x > 7 \end{cases}$$

```
import math
x = float(input("Ievadi skaitli ==>"))
if (x>=-2*math.pi) and (x<=2*math.pi) :
    y=x*math.sin(x)
    print("F(",str(x),")=",str(y))
elif x<-7 :
    y=abs(x)
    print("F(",str(x),")=",str(y))
elif x>7 :
    y=x-7
    print("F(",str(x),")=",str(y))
else:
    print("Nav definēts")
```



Trīsstūra tipa noteikšana + laukums

Vai var izveidot?

$$A < B + C$$

$$B < A + C$$

$$C < B + C$$

Ja C ir lielākā, tad pietiek $C < A + B$

Tips pēc leņķa

Pret lielākās malas atrodas lielākais leņķis

Pret lielākā leņķa atrodas lielākā mala

$$C^2 = A^2 + B^2 - 2 \cdot A \cdot B \cdot \cos \gamma$$

Tips pēc leņķa

Pret lielākās malas atrodas lielākais leņķis

Pret lielākā leņķa atrodas lielākā mala

$$C^2 = A^2 + B^2 - 2 \cdot A \cdot B \cdot \cos \gamma$$

Taisnleņķa $C^2 = A^2 + B^2$

Šaurleņķa $C^2 < A^2 + B^2$

Platleņķa $C^2 > A^2 + B^2$

Tips pēc malu garumiem

$A=B$ $B=C$ $A=C$,

ievērojot transitīvo īpašību pietiek $A=B$ $B=C$

$A=B$ vai $B=C$ vai $A=C$

$A<>B$ un $B<>C$ vai $A<>C$

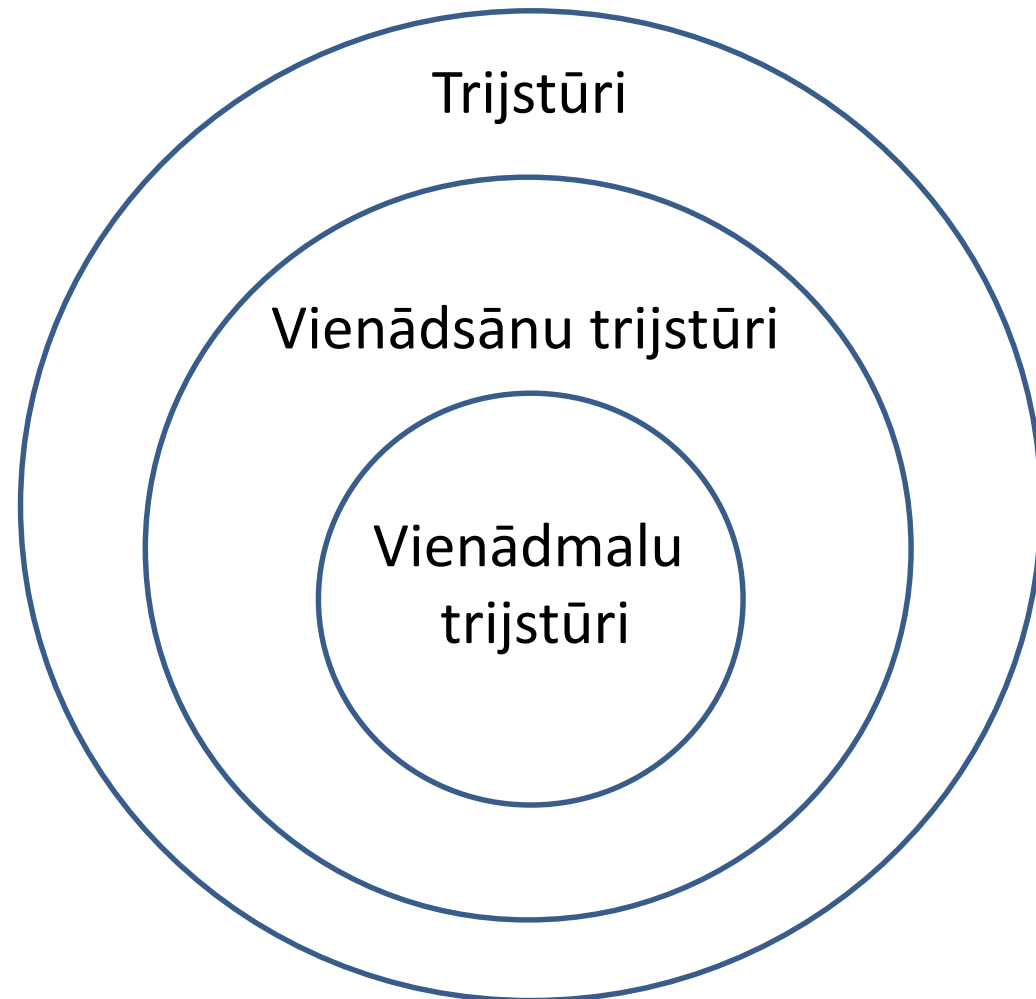
Tips pēc malu garumiem

$A=B$ $B=C$ $A=C$,

ievērojot transitīvo īpašību pietiek $A=B$ $B=C$

$A=B$ vai $B=C$ vai $A=C$

$A \neq B$ un $B \neq C$ vai $A \neq C$



```
import math
```

```
a = float(input("Ievadi 1. malu ==>"))
```

```
b = float(input("Ievadi 2. malu ==>"))
```

```
c = float(input("Ievadi 3. malu ==>"))
```

```
# garākā mala būs c
```

```
if c < a :
```

```
    x=c
```

```
    c=a
```

```
    a=x
```

```
if b>c :
```

```
    x=b
```

```
    b=c
```

```
    c=x
```

```
if a+b<=c :
    print("malas neveido trijsturi")
else :
    # lenķis
    if c*c==a*a+b*b :
        y = "taisnlenķa "
    elif c*c>=a*a+b*b :
        y = "platlenķa "
    else :
        y = "šaurlenķa "
    # malas
    if (a==b) and (b==c) :
        z = "vienādmalu "
    elif (a==b) or (b==c) or (c==a) :
        z= "vienādsānu "
    else :
        z= "dažādmalu "
```

```
# laukums
p=(a+b+c) /2
s=math.sqrt(p*(p-a)*(p-b)*(p-c))

print("Tas ir " + y + z + "trijstūris")
print("Tā laukums ir: ", s)
```

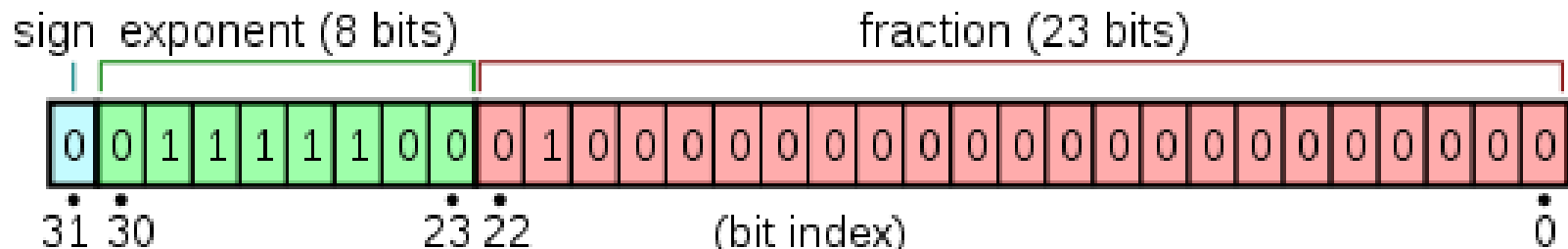
Datu tipi un aprēķinu rezultātu precizitāte

- Veselie skatļi
- Skaitļi ar *peldošo* komatu

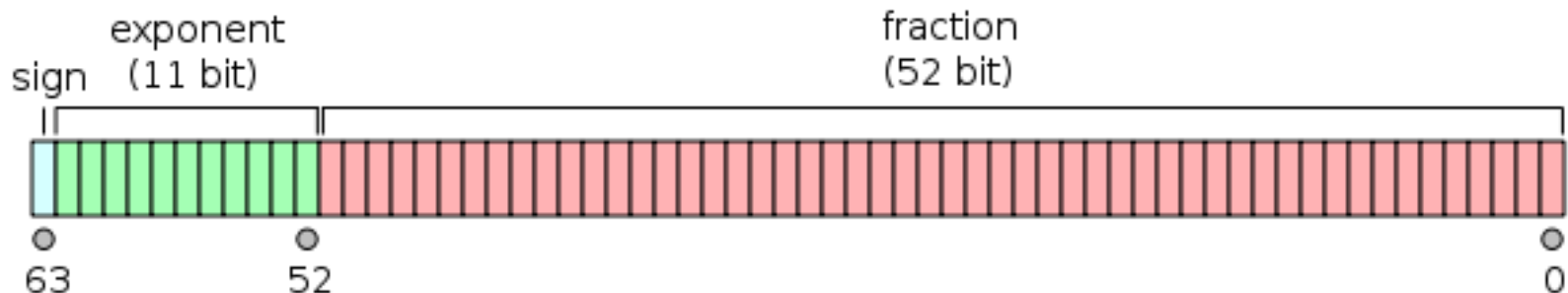
Skaitļi ar *peldošo* komatu

Elektronikas un Elektrotehnikas inženieru institūts (IEEE) ir standartizējis peldošā punkta reprezentācijas standartā IEEE 754, kam seko vairums mūsdienu datoru (*dažādi varianti*):

- Vienkārša precizitāte (*single precision*) - 32 biti jeb 4 baiti



- Dubultprecizitāte (*double precision*) – 64 biti jeb 8 baiti



```
a = 888888888888888887
b = 999999999999999998
c = 8
d = 9
```

```
d1 = a / b
d2 = c / d
```

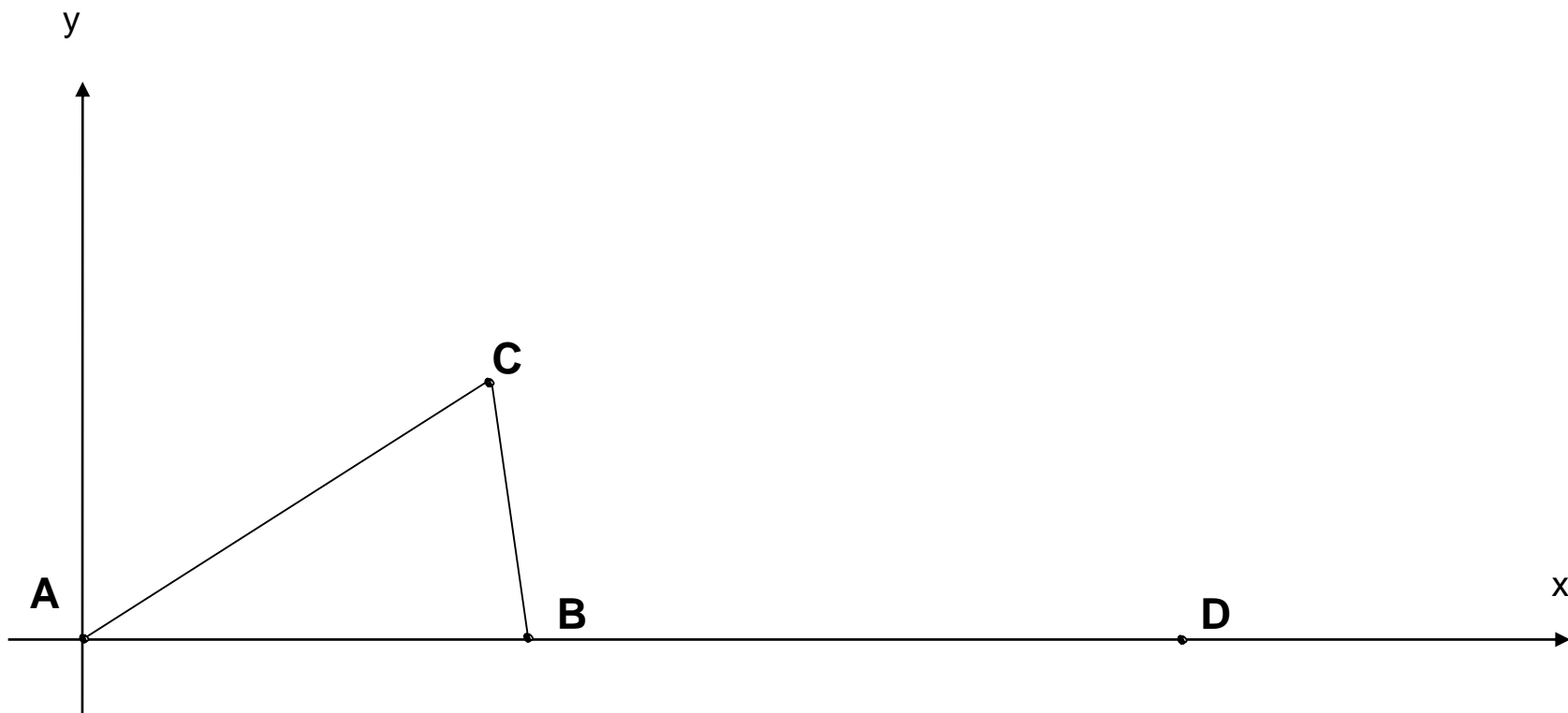
```
r1=a*d
r2=b*c
```

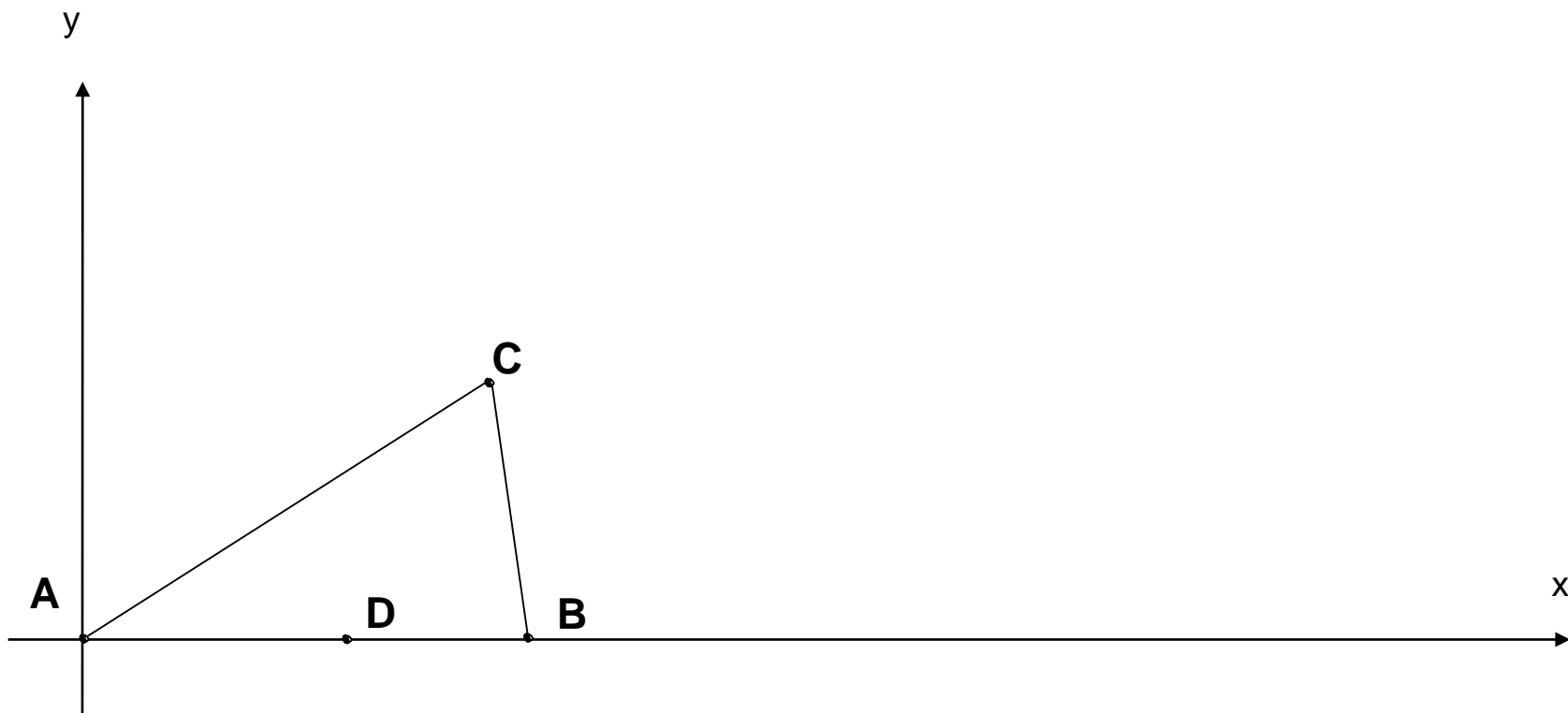
$A : B = C : D$

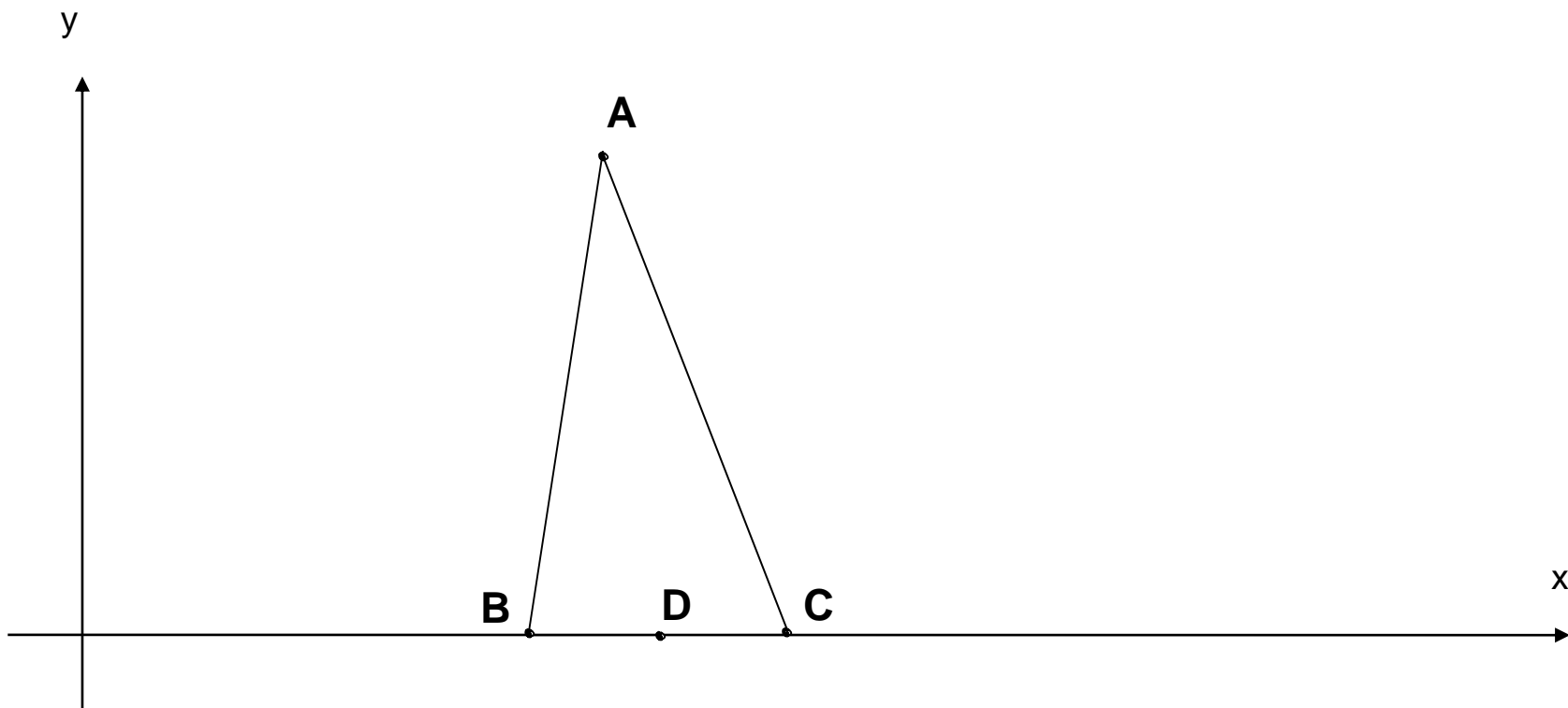
$A \cdot D = B \cdot C$

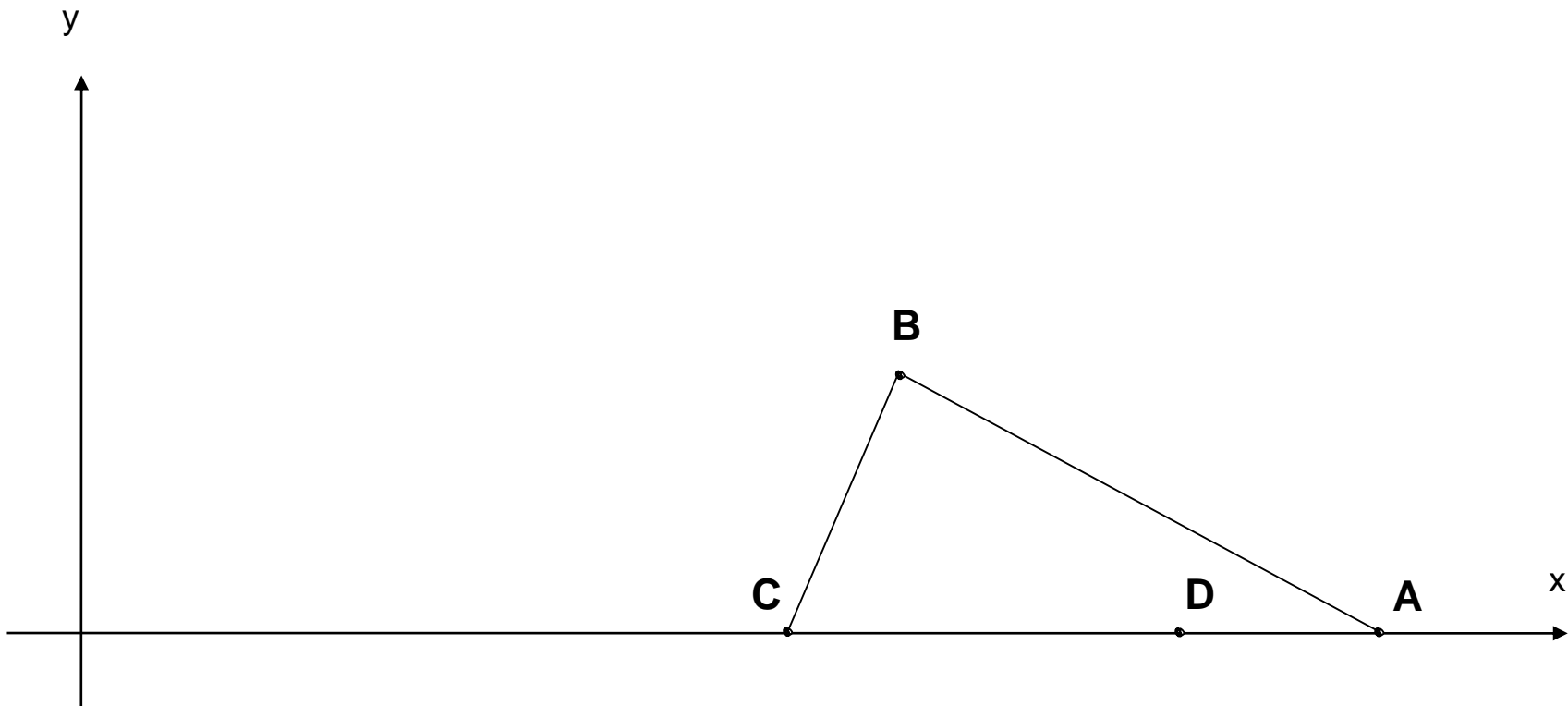
```
# pirmā metode
print(d1)
print(d2)
if d1==d2 :
    print ("Ir vienādi")
else:
    print ("Nav vienādi")
```

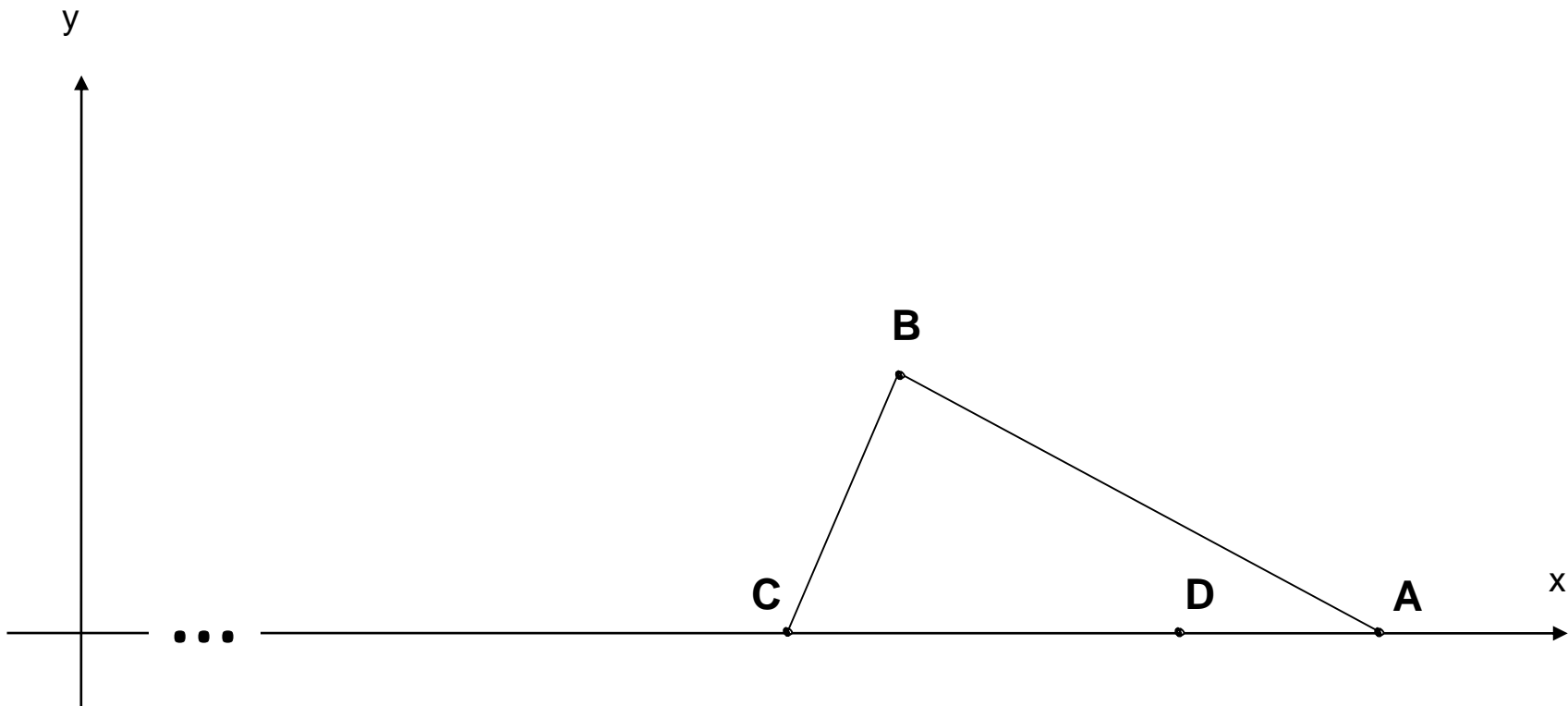
```
# otrā metode
print(r1)
print(r2)
if r1==r2 :
    print ("Ir vienādi")
else:
    print ("Nav vienādi")
```

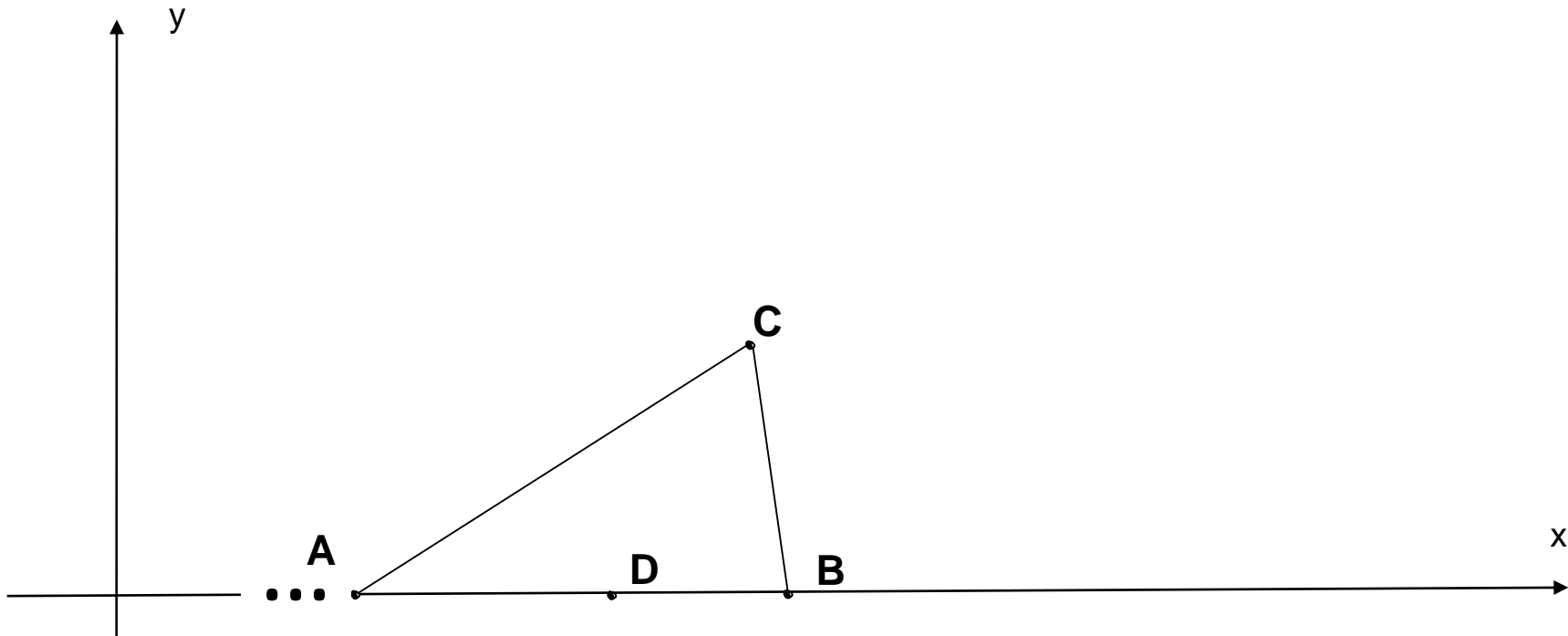


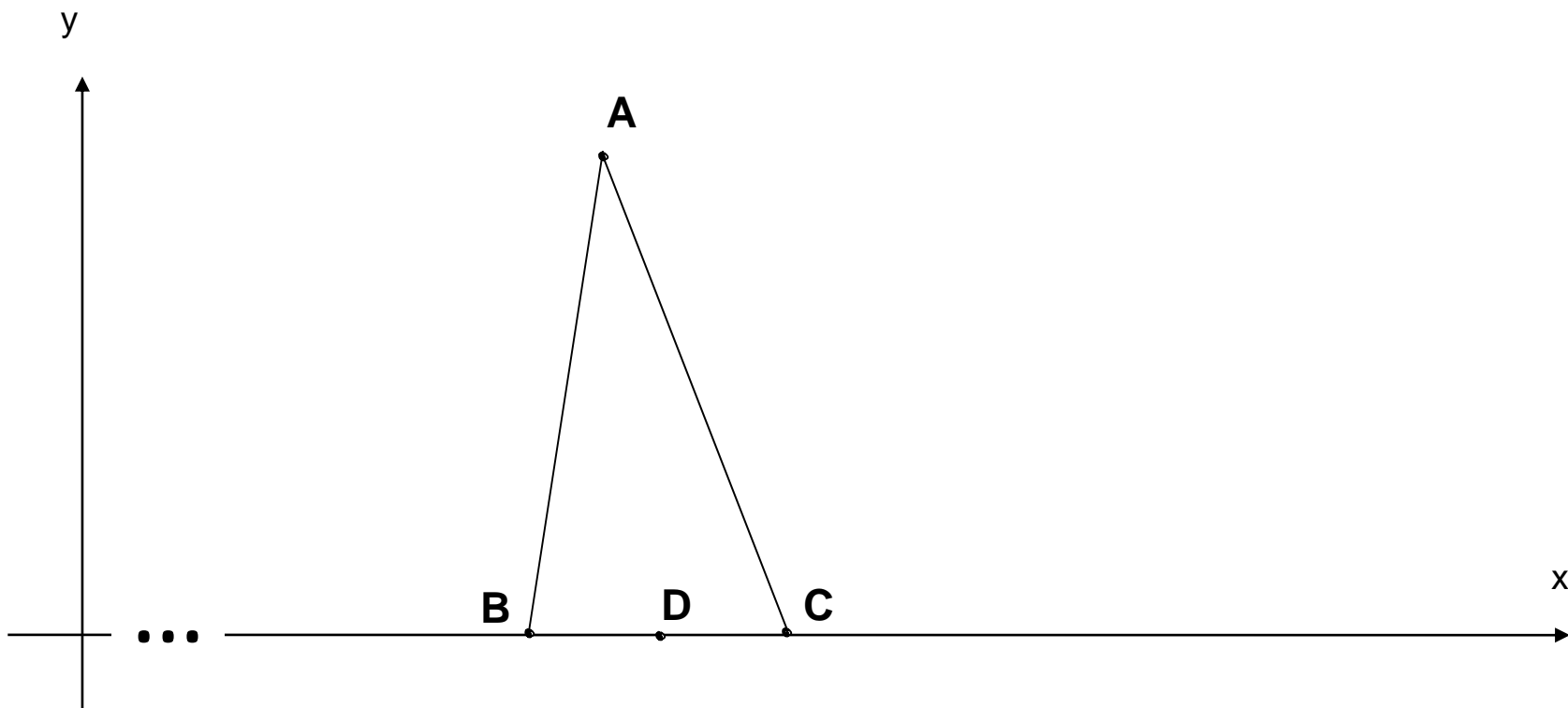


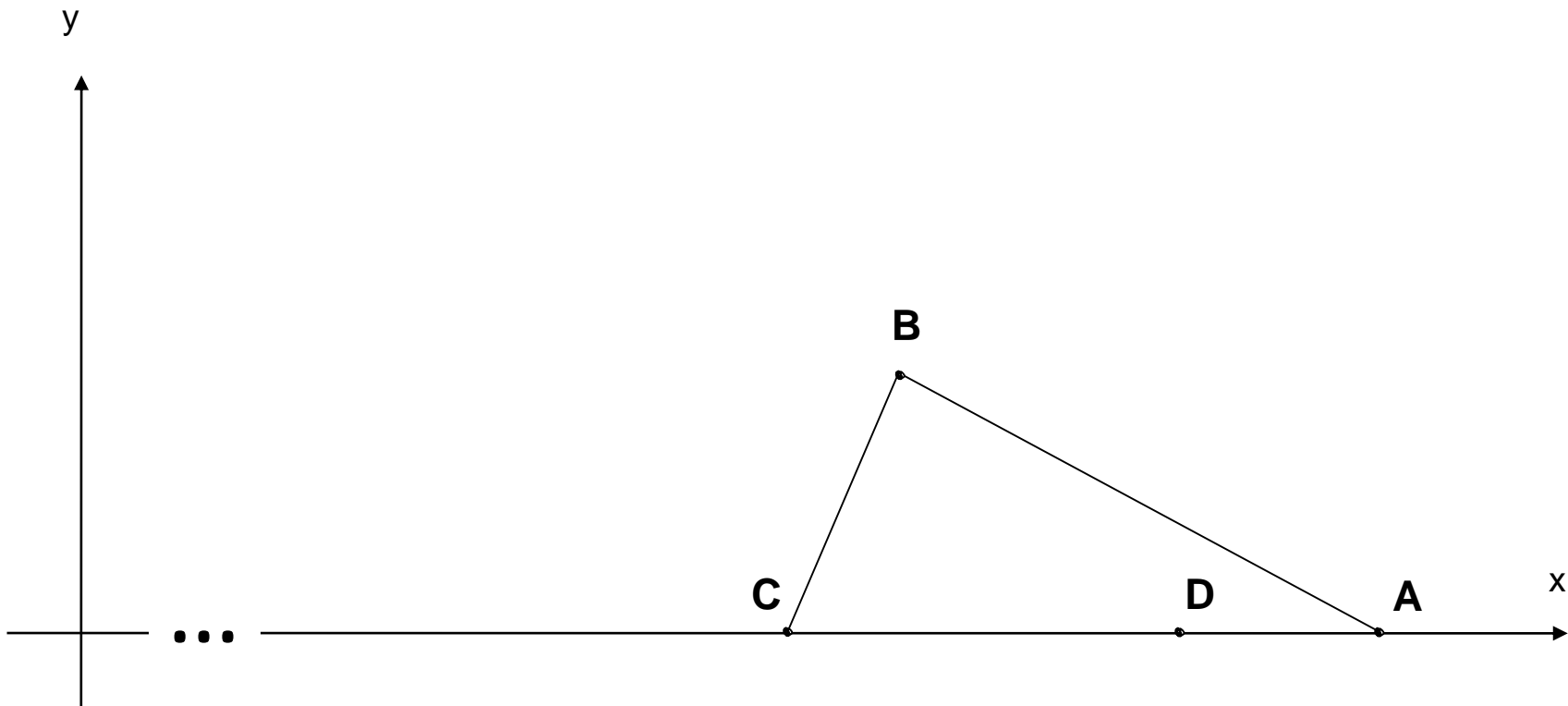












```
import math
```

```
x = float(input("Ievadi X ==> "))
```

```
y = float(input("Ievadi Y ==> "))
```

```
z = float(input("Ievadi Z ==> "))
```

```
d = float(input("Ievadi D ==> "))
```

```
# aprēķini
```

```
ab = x
```

```
bc = math.sqrt((x - y) * (x - y) + z * z)
```

```
ac = math.sqrt(y * y + z * z)
```

```
p = ab + bc + ac
```

```
ee = d - p * math.trunc(d / p)
```



```
# kas uzkrita
if ee == 0 :
    vv = "virсотne A"
elif ee < ab :
    vv = "mala AB"
elif ee == ab :
    vv = "virсотne B"
elif ee < (ab + bc) :
    vv = "mala BC"
elif ee == (ac + bc) :
    vv = "virсотne C"
else :
    vv = "mala AC"

print("Uz puntu D uzkrita " + vv)
```




A diagram illustrating a line in a 2D coordinate system. A solid black line slopes downwards from left to right. Two points are marked on the line: point E is located higher and further to the left, while point D is located lower and further to the right. To the left of the line, the equation $Ax + By + C = 0$ is written.

$$Ax + By + C = 0$$

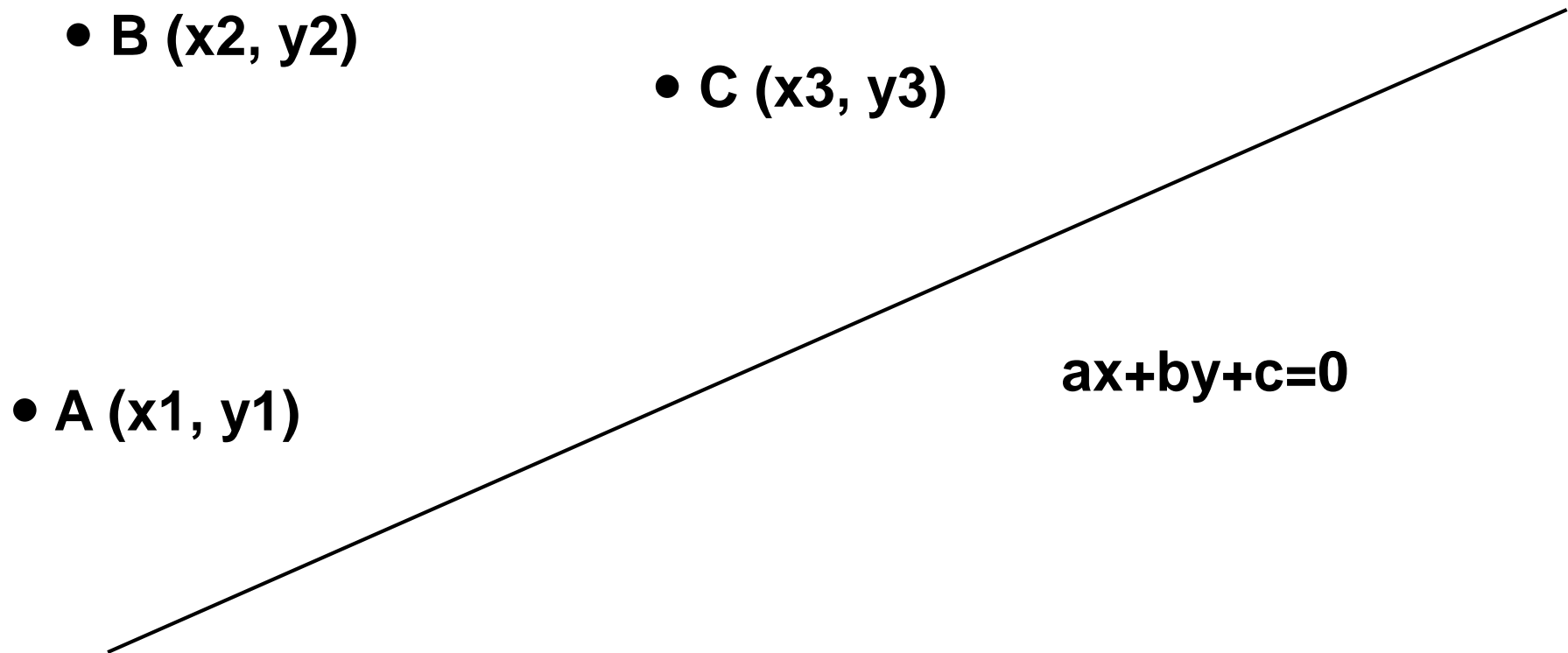
• $E(x_1, y_1)$

• $D(x_2, y_2)$

```
a = float(input("Ievadi A ==> "))
b = float(input("Ievadi B ==> "))
c = float(input("Ievadi C ==> "))
x1 = float(input("Ievadi X1 ==> "))
y1 = float(input("Ievadi Y1 ==> "))
x2 = float(input("Ievadi X2 ==> "))
y2 = float(input("Ievadi Y2 ==> "))
```

```
z1=a*x1+b*y1+c
z2=a*x2+b*y2+c
```

```
if z1*z2>0 :
    print ("Punkti ir vienā pusē taisnei.")
else :
    print ("Punkti nav vienā pusē taisnei.")
```



```
a = float(input("Ievadi A ==> "))
b = float(input("Ievadi B ==> "))
c = float(input("Ievadi C ==> "))
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 ==> "))
```

```
z1=a*x1+b*y1+c
```

```
z2=a*x2+b*y2+c
```

```
z3=a*x3+b*y3+c
```

```
if z1*z2*x1>0 :
```

```
    print ("Punkti ir vienā pusē taisnei.")
```

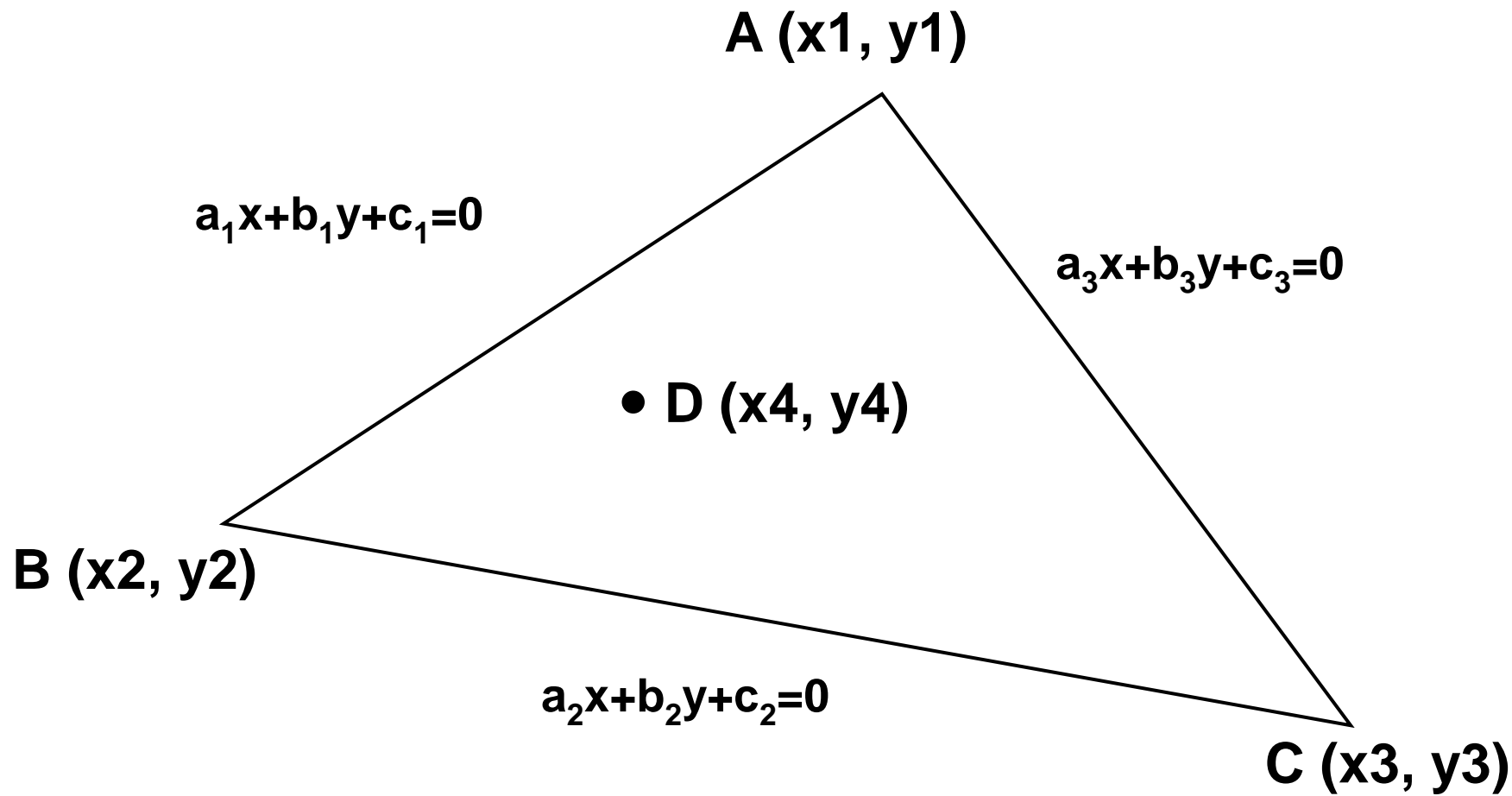
```
else :
```

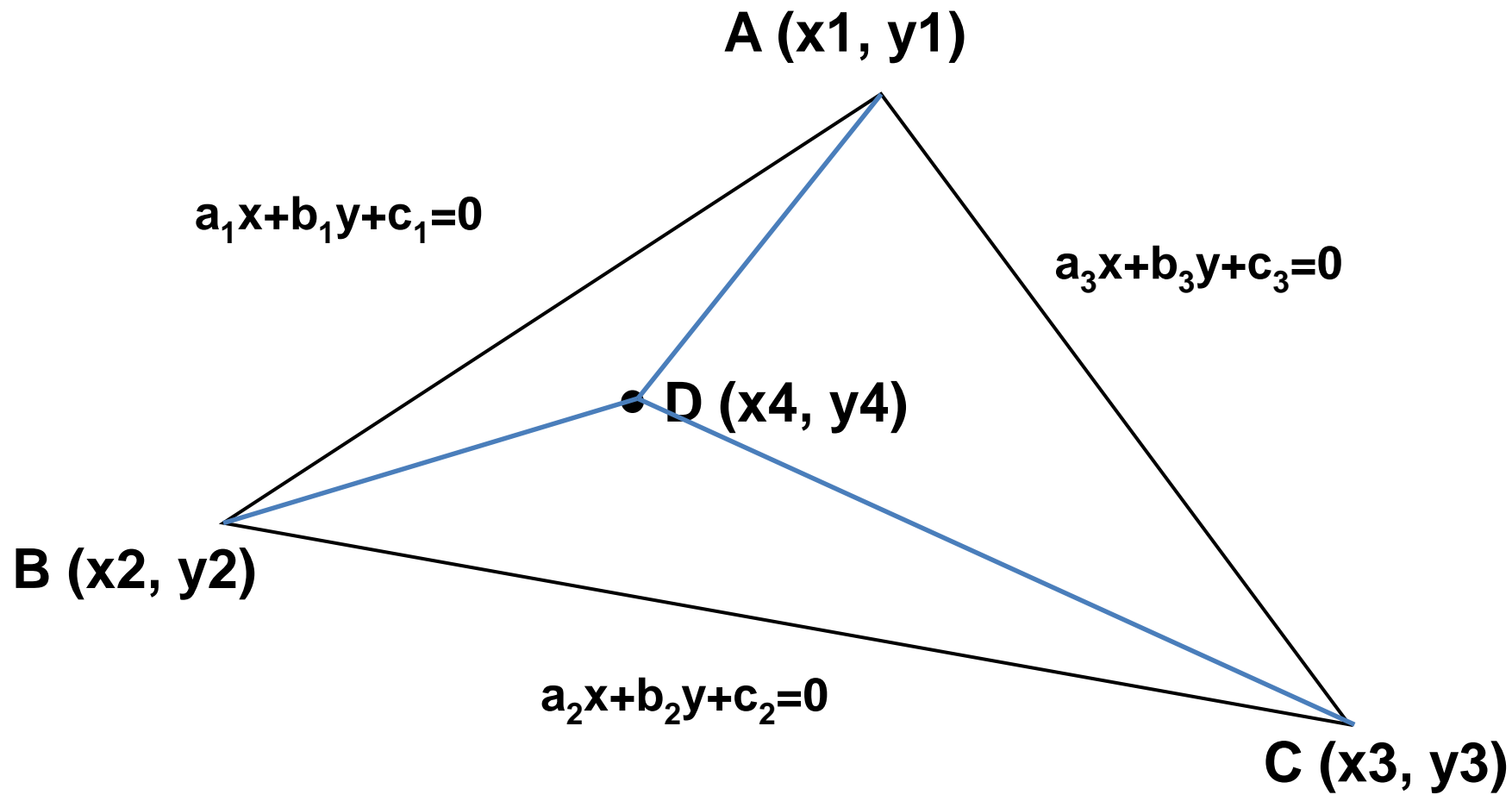
```
    print ("Punkti nav vienā pusē taisnei.")
```

```
a = float(input("Ievadi A ==> "))
b = float(input("Ievadi B ==> "))
c = float(input("Ievadi C ==> "))
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 ==> "))
```

```
z1=a*x1+b*y1+c
z2=a*x2+b*y2+c
z3=a*x3+b*y3+c
```

```
if (z1*z2>0) and (z1*z3>0) :
    print ("Punkti ir vienā pusē taisnei.")
else :
    print ("Punkti nav vienā pusē taisnei.")
```





Datu ievade

X1= 0 Y1= 0

X2= 2 Y2= 0

X3= 0 Y3= 2

X4= 1 Y4= 1

Rezultāts

S= 2.000

S1= 1.000

S2= 0.500

S3= 0.500

1.metode => Punkts D neatrodas trijstūra iekšpusē

2.metode => Punkts D atrodas trijstūra iekšpusē

Aprēķināt!

Beigt

```
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))
```

```
# aprēķini 1.veids
```

```
p = (ab + bc + ac) / 2
```

```
s = math.sqrt(p * (p - ab) * (p - bc) * (p - ac))
```

```
p = (ab + ad + bd) / 2
```

```
s1 = math.sqrt(p * (p - ab) * (p - ad) * (p - bd))
```

```
p = (bc + bd + cd) / 2
```

```
s2 = math.sqrt(p * (p - bc) * (p - bd) * (p - cd))
```

```
p = (ac + ad + cd) / 2
```

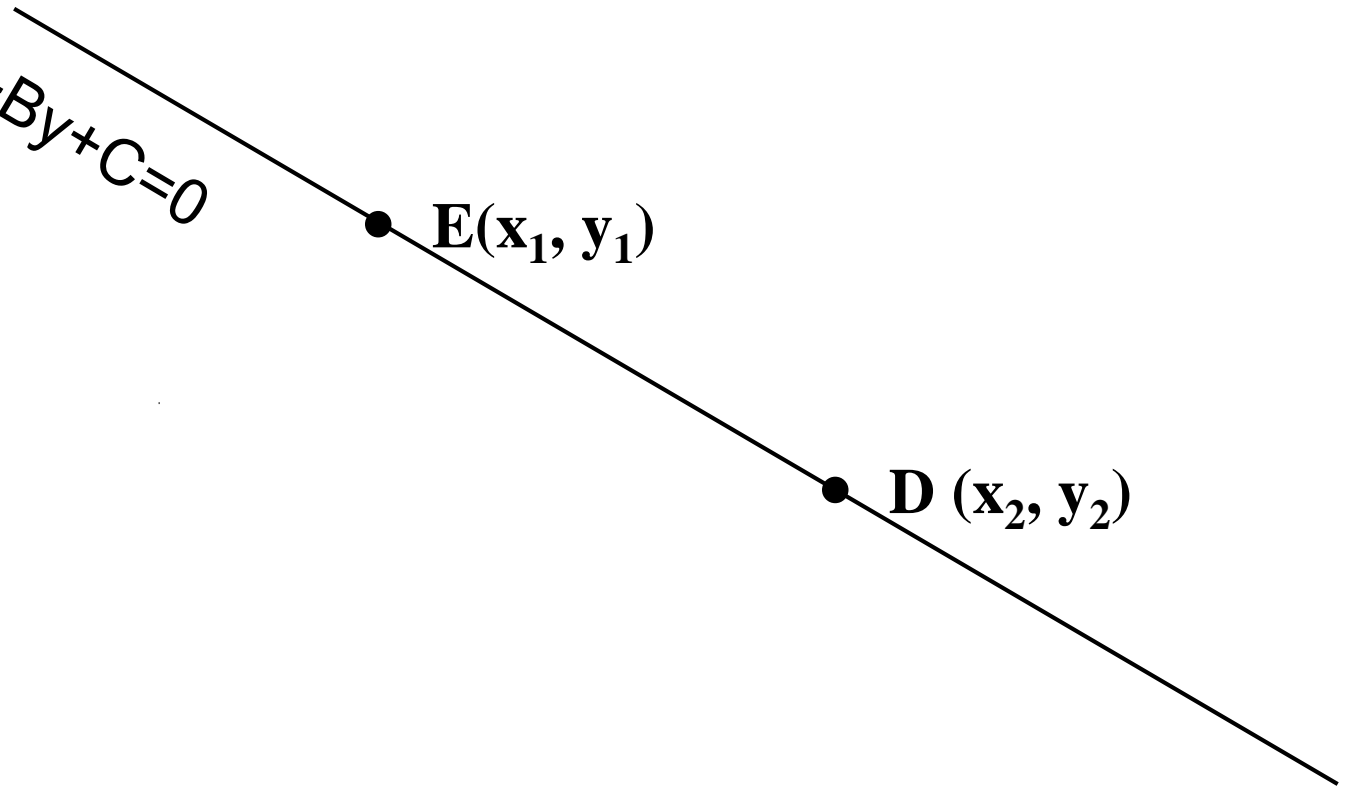
```
s3 = math.sqrt(p * (p - ac) * (p - ad) * (p - cd))
```

```
if s == s1 + s2 + s3 :
```

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

```
else :
```

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

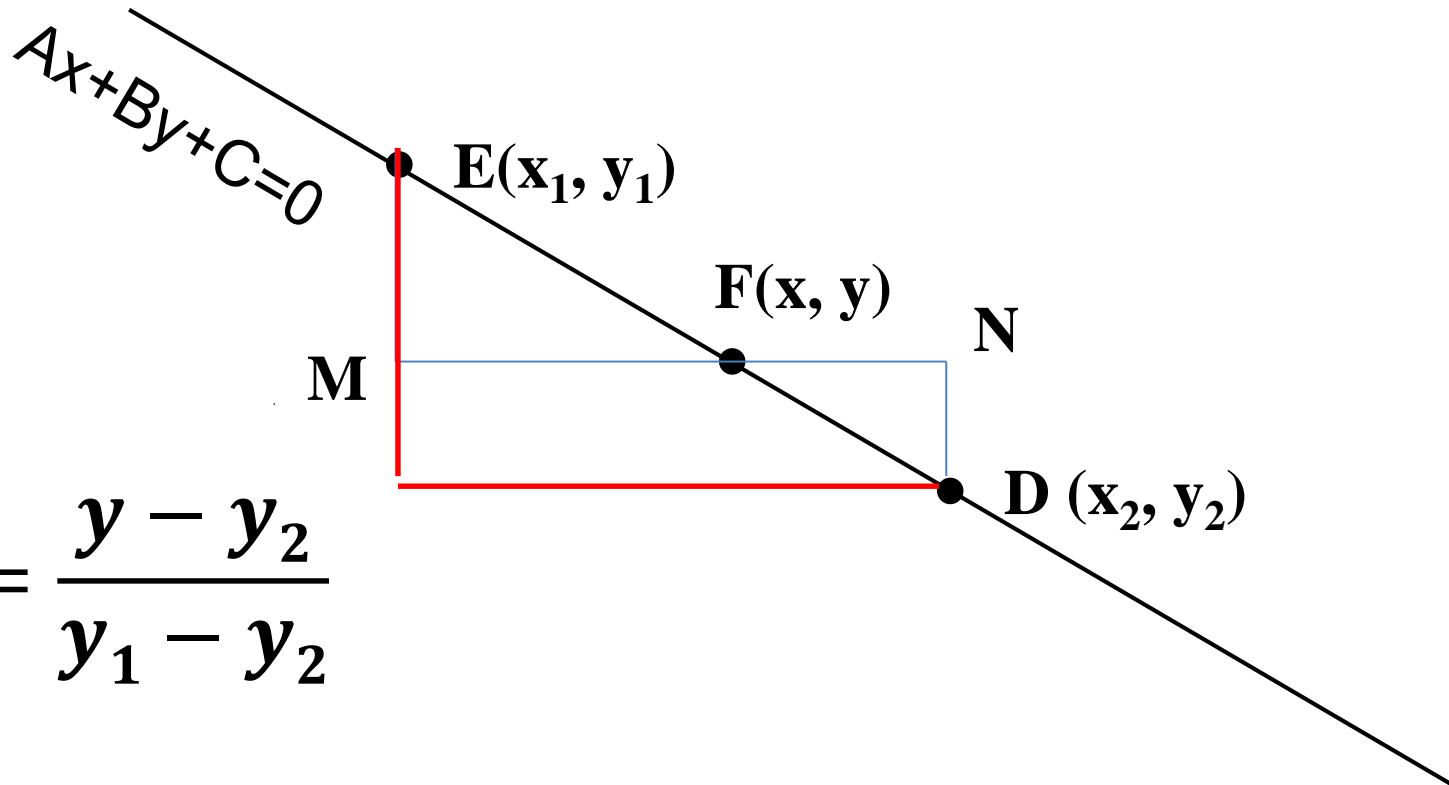


A diagram illustrating a line in a 2D coordinate system. The line is represented by a solid black line with a negative slope. Two points, labeled E and D, are marked on the line with black dots. Point E is located at a higher position on the line than point D. The equation of the line, $Ax+By+C=0$, is written in the upper left corner. The coordinates of the points are given as $E(x_1, y_1)$ and $D(x_2, y_2)$.

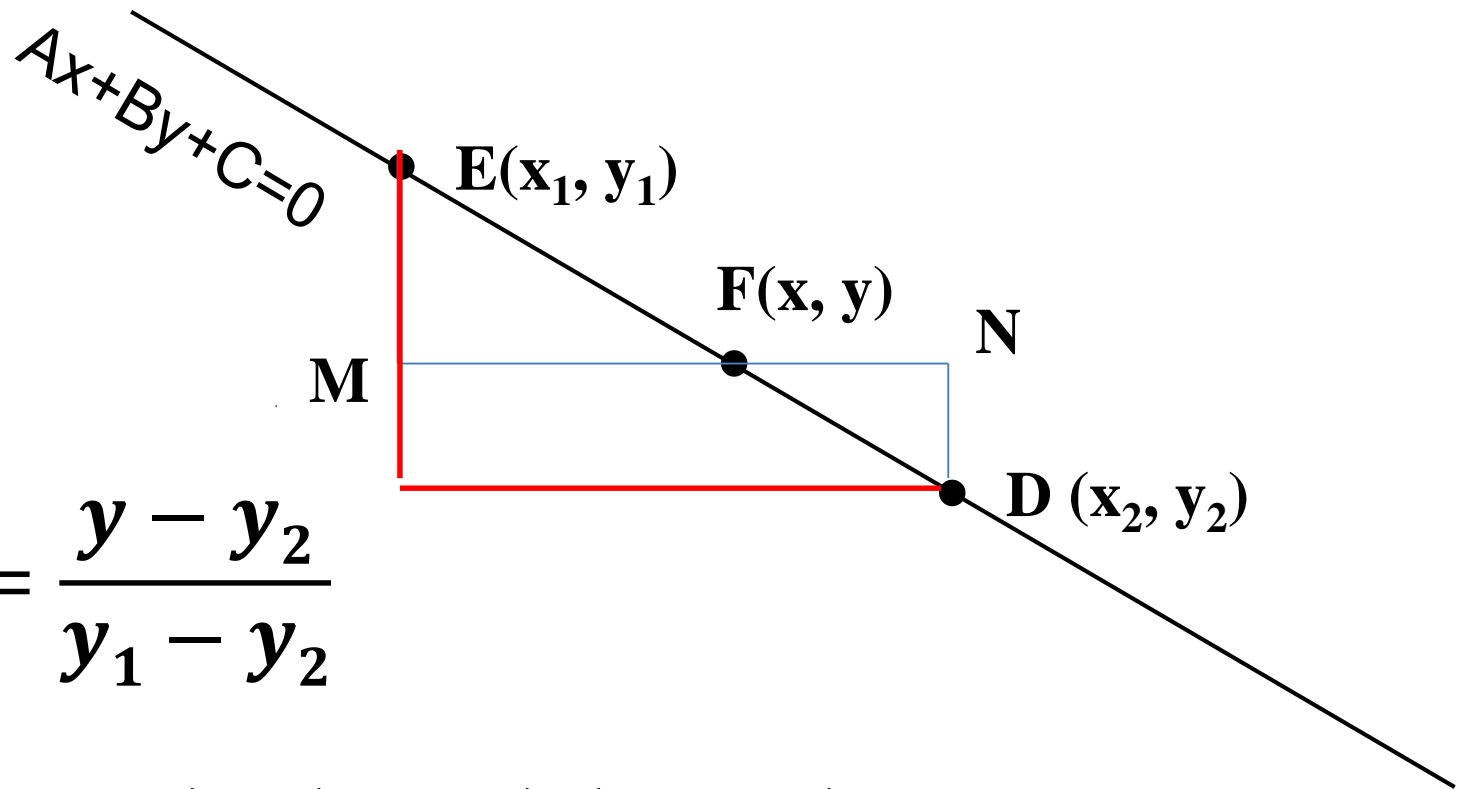
$$Ax+By+C=0$$

$E(x_1, y_1)$

$D(x_2, y_2)$



$$\frac{x - x_2}{x_1 - x_2} = \frac{y - y_2}{y_1 - y_2}$$



$$\frac{x - x_2}{x_1 - x_2} = \frac{y - y_2}{y_1 - y_2}$$

$$(x - x_2)(y_1 - y_2) = (y - y_2)(x_1 - x_2)$$

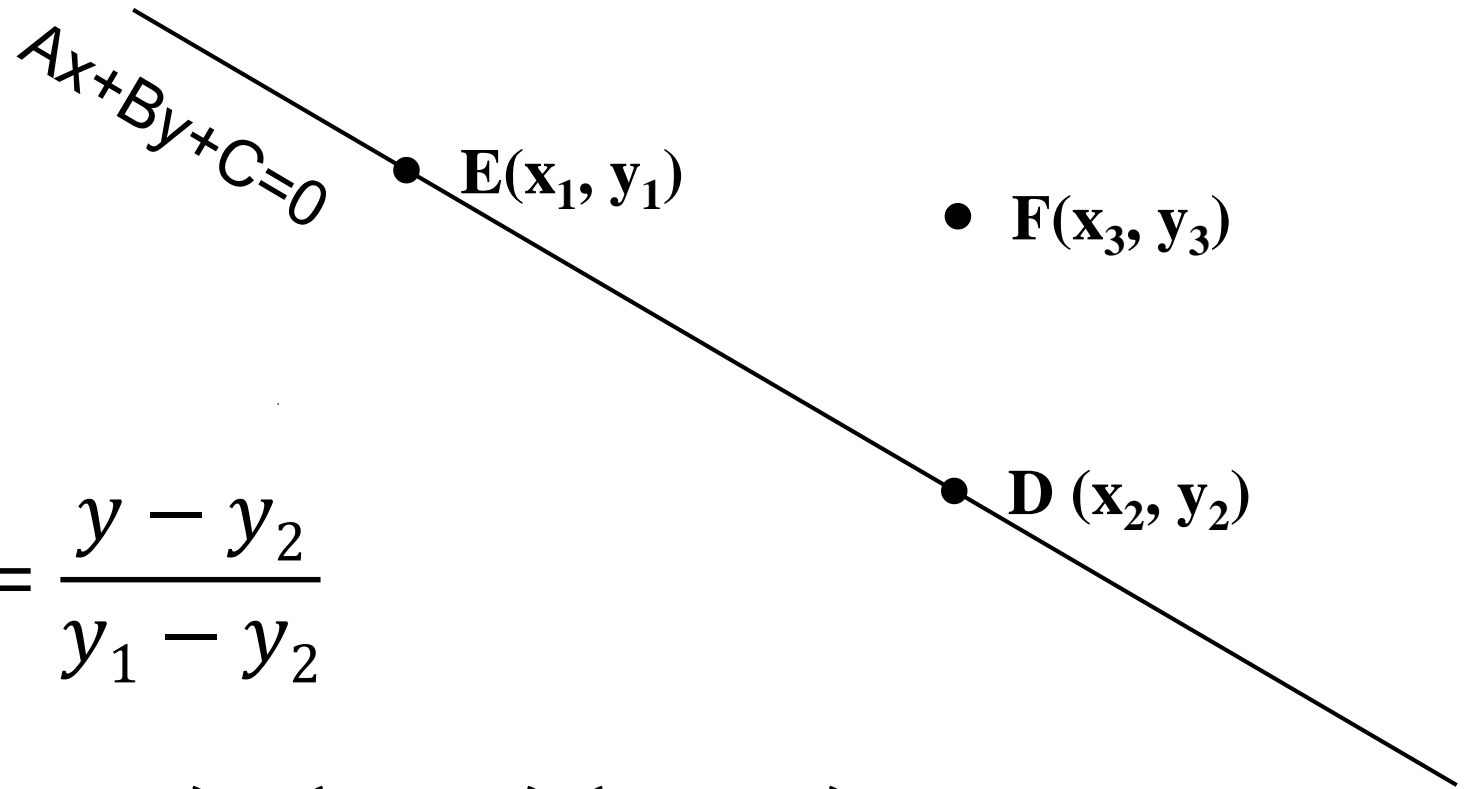
$$(x - x_2)(y_1 - y_2) - (y - y_2)(x_1 - x_2) = 0$$

$$\underbrace{(y_1 - y_2)x}_{A} - \underbrace{(x_1 - x_2)y}_{B} - \underbrace{x_2(y_1 - y_2) + y_2(x_1 - x_2)}_{C} = 0$$

A

B

C



$$\frac{x - x_2}{x_1 - x_2} = \frac{y - y_2}{y_1 - y_2}$$

$$(x - x_2)(y_1 - y_2) = (y - y_2)(x_1 - x_2)$$

$$(x - x_2)(y_1 - y_2) - (y - y_2)(x_1 - x_2) = 0$$

$$(x_3 - x_2)(y_1 - y_2) - (y_3 - y_2)(x_1 - x_2) = 0$$

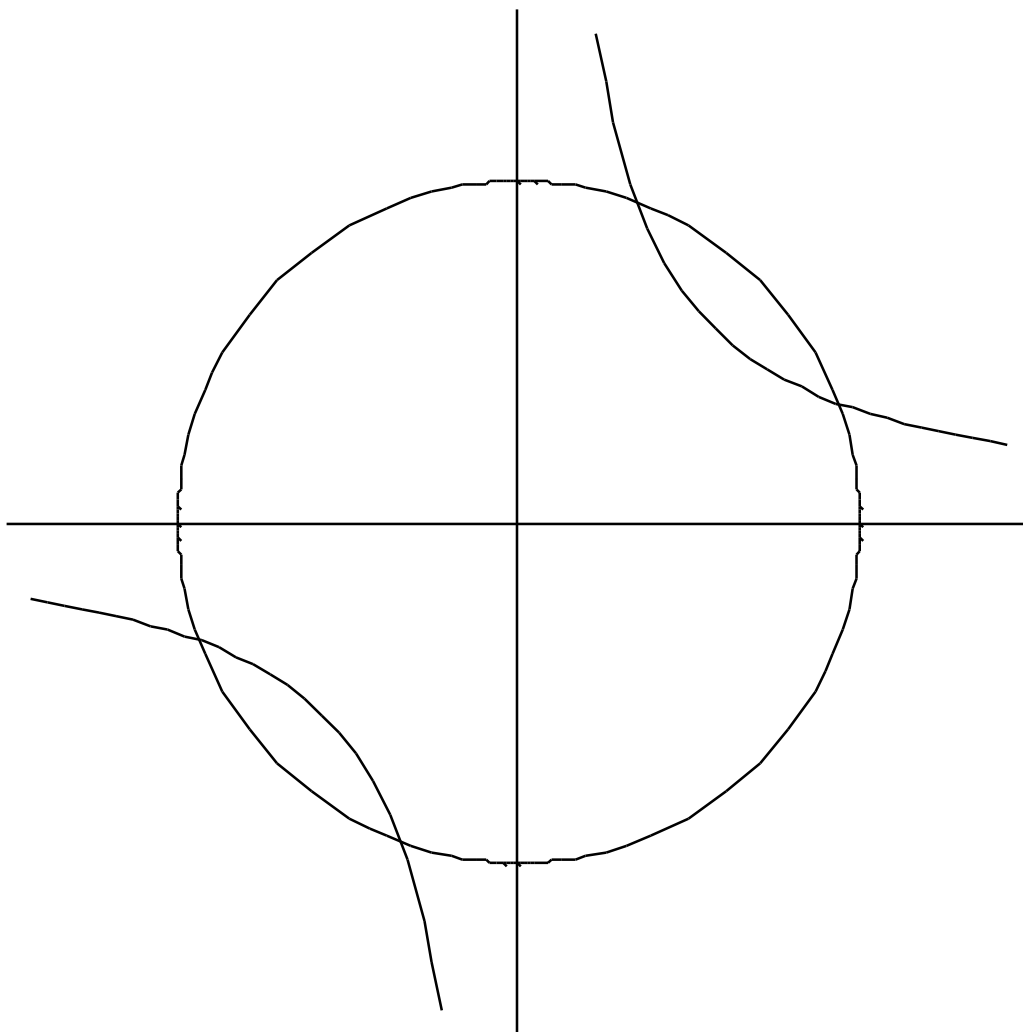
aprēķini 2.veids

```
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")
```

1. domājamais uzdevums

$$\begin{cases} Ax^2 + Ay^2 = B \\ Cxy = D \end{cases}$$



2. domājamais uzdevums

$$\begin{cases} ax^2 + by^2 + c = 0 \\ dx^2 + gy + f = 0 \end{cases}$$

