**Ieskaite. Atkļūdošana, programmas veidošana pēc dotas specifikācijas.**

**Iesniegt arī daļējus risinājumus**

1. Galvenē nomainīt Vards un Uzvards ar savu vārdu un uzvārdu. Tabulas pirmajā rindiņā ierakstiet sava personas koda pirmo daļu. Šos datus izmantos tālākajos uzdevumos (1 p.)

| 0 | 8 | 0 | 0 | 0 | 4 |
| --- | --- | --- | --- | --- | --- |
| **a** | **b** | **c** | **d** | **e** | **f** |

2. Kļūdu veidi. Uzrakstiet vienu piemēru katram minētajam kļūdas veidam, paskaidrot kļūdas rašanās iemeslu (6 p.)

a: **izpildlaika kļūda**

Kods ir pareizs, bet norāditais fails nav atrasts

file = open('text.txt').read().split()

print(file)

b: **sintakses kļūda**

Kodā ir ierakstīts lieks simbols

file = open('text.txt').read().split()a

print(file)

c: **loģikas kļūda**

Kodā ir aprēķināšanas kļūda

Perimiter = (a+a+a)

3. Ir teksta fails *Dati\_in.txt*. Tajā, trīs rindiņās pēc kārtas, ir dotas 3 punktu A, B un C koordinātes. Katra punkta koordinātes ir savā rindiņā, x un y vērtības ir atdalītas ar vienu atstarpīti. Teksta failā *Dati\_out.txt* pirmajā rindiņā tiek izvadīts komentārs “Trīsstūri nevar izveidot”, ja dotajiem ievaddatiem trīsstūri nevar izveidot un komentāru “Trīsstūri var izveidot” pretējā gadījumā. Nākamajās 3 rindās izvadīt trīsstūra malu garumus ar precizitāti 2 cipari aiz komata. Paraugs:

*Mala a = 34.00*

*Mala b = 345.02*

*Mala c = 0.01*

Nākamajās divās rindās ir izvadītas trīsstūra perimetra un laukuma vērtības šādā formātā:

*Perimetrs P = 456.00 gv*

*Laukums S = 653.02 kv*

Caur virsotni A paralēli malai [BC] ir novilkta taisne. Ir noteikts šīs taisnes vienādojums un izvadīts nākamajā rindā pēc laukuma vērtības šādā formātā:

*y = 34.45\*x – 23.34*

Abas skaitliskās vērtības var būt citas, atkarībā no ievaddatiem.

Šeit iekopēt programmas tekstu: (20 p.)

import math

import linecache

#Punkts A

x1 = float(linecache.getline("dati\_in.txt", 2))

y1 = float(linecache.getline("dati\_in.txt", 4))

#Punkts B

x2 = float(linecache.getline("dati\_in.txt", 6))

y2 = float(linecache.getline("dati\_in.txt", 8))

#Punkts C

x3 = float(linecache.getline("dati\_in.txt", 10))

y3 = float(linecache.getline("dati\_in.txt", 12))

#a^2 + b^2 = c^2

ax = (x2-x3)\*\*2 + (y2-y3)\*\*2

bx = (x1-x3)\*\*2 + (y3-y1)\*\*2

cx = (x1-x2)\*\*2 + (y2-y1)\*\*2

#Pārbauda vai trīssturis ir iespējams

if ax <= 0 or bx <= 0 or cx <= 0:

file = open("dati\_out.txt", "w")

file.write("Trīsstūri nevar izveidot.")

file.close()

else:

#Malu garumi iegūti no kvadrātsaknēm

a = math.sqrt(ax)

b = math.sqrt(bx)

c = math.sqrt(cx)

Perimiter = (a+b+c)

#Laukuma iegūšana

s = (a + b + c) / 2

Area = (s\*(s-a)\*(s-b)\*(s-c)) \*\* 0.5

#Aprēķina m ar formulu m = "(y2-y1) / (x2 - x1)"

m = (x3 - y3) / (x2 - y2)

#Aprēķina t ar forumlu "y = mx + t", nav nepieciešams šajā piemērā

#t = -x3 \* m + y3

#Aprēķina t no punkta A ar formulu "y = mx + t"

ta = -(m\*x1)+y1

file = open("dati\_out.txt", "w")

file.write("Trīsstūri var izveidot." + "\n")

file.write("Mala a = " + str(a) + "\n")

file.write("Mala b = " + str(b) + "\n")

file.write("Mala c = " + str(c) + "\n")

file.write("Perimetrs P = " + str(Perimiter) + "\n")

file.write("Laukums S = " + str(Area) + "\n")

file.write("y = " + str(m) + "x + " + str(ta))

file.close()

4. Izstrādāt 3. uzdevumā izveidotajai programmai testa plānu (10 p.)

Tabulu ar testa plānu iekopēt šeit:

| Testa Nr. | Testa Tips | Ievadītie Dati | Testa Mērķis | Sagaidāmais rezūltāts | Rezūltāts | Tests izturēts/neizturēts |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Derīgs | x1=1,y1=1,x2=6,y2=1,x3=7,y3=1 | Pārbaudīt vai iespējams trīssturis tiks identificēts kā iespējams, pareizi aprēķināts un izvadīts | Trīsstūri var izveidot  Mala a = 7.810249675906654  Mala b = 6.0  Mala c = 5.0  Perimetrs P = 18.810249675906654  Laukums S = 15.0  y = -1.2x + 2.2 | derīgs | izturēts |
| 2 | Nederīgs | x1=0,y2=8,x2=0,y2=8,x3=0,y3=4 | Pārbaudīt vai neiespējams trīsturis tiks identificēts un izvadīts | Trīssturi nevar izveidot | derīgs | izturēts |

Papildus testēšānu var veikt ar <https://www.geogebra.org/>

| Balles | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Punkti | 0 - 3,8 | 3,8 - 5,7 | 5,7 - 9,5 | 9,5 - 11,4 | 11,4 - 13,3 | 13,3 - 14,2 | 14,3 - 15,2 | 15,2 - 16,1 | 16,2 - 18 | 18,1 - 19 |