

1. uzdevums

Izveidot programmu, kurā ievadot divus veselus skaitļus, tiek iegūta šo skaitļu summa (+), starpība (-), reizinājums (*), dalījums (/), vesels dalījums (//) un dalīšanas atlikums (%).

Kods:

```
# Programmas nosaukums: Dažas operācijas ar diviem veseliem skaitļiem
```

```
# 1. uzdevums
```

```
# Uzdevuma formulējums: Izveidot programmu, kurā ievadot divus veselus skaitļus, tiek iegūta šo skaitļu summa (+), starpība (-), reizinājums (*), dalījums (/), vesels dalījums (//) un dalīšanas atlikums (%).
```

```
# Programmas autors: Vladislavs Babanins
```

```
# Versija 1.4
```

```
print("\nDažas operācijas ar diviem veseliem skaitļiem\n\nVersija 1.4\n") # Programmas nosaukums un versija parādas
```

```
a=int(input("Ievadi pirmo veselo skaitli ==>")) # Pieprasa skaitli a un input'u => integer
```

```
b=int(input("Ievadi otro veselo skaitli ==>")) # Pieprasa skaitli b un input'u => integer
```

```
print(str(a) + "+" + str(b) + "=" + str(a+b) + " Summa") # Vienā rinda programma saskaita a un b un ievāda to tekstā str() veidā
```

```
print(str(a) + "-" + str(b) + "=" + str(a-b) + " Starpība") # Pēc analogijas izdarīts tas pats visām nepieciešamām operācijām
```

```
print(str(b) + "-" + str(a) + "=" + str(b-a) + " Starpība")
```

```
print(str(a) + "*" + str(b) + "=" + str(a*b) + " Reizinājums")
```

```
print(str(a) + "/" + str(b) + "=" + str(a/b) + " Dalījums")
```

```
print(str(b) + "/" + str(a) + "=" + str(b/a) + " Dalījums")
```

```
print(str(a) + "//" + str(b) + "=" + str(a//b) + " Veselais dalījums")
```

```
print(str(b) + "//" + str(a) + "=" + str(b//a) + " Veselais dalījums")
```

```
print(str(a) + "%" + str(b) + "=" + str(a%b) + " Dalīšanas atlikums")
```

```
print(str(b) + "%" + str(a) + "=" + str(b%a) + " Dalīšanas atlikums")
```

Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Dažas operācijas ar diviem veseliem skaitļiem
# 1. uzdevums
# Uzdevuma formulējums: Izveidot programmu, kurā ievadot divus veselus skaitļus, tiek iegūta šo skaitļu summa (+), starpība (-), reizinājums (
# Programmas autors: Vladislavs Babajins
# Versija 1.4

print("\nDažas operācijas ar diviem veseliem skaitļiem\n\nVersija 1.4\n") # Programmas nosaukums un versija parādas

a=int(input("Ievadi pirmo veselo skaitli ==>")) # Pieprasa skaitli a un input'u => integer
b=int(input("Ievadi otro veselo skaitli ==>")) # Pieprasa skaitli b un input'u => integer

print(str(a) + "+" + str(b) + "=" + str(a+b) + " Summa") # Vienā rinda programma saskaita a un b un ievāda to tekstā str() veidā
print(str(a) + "-" + str(b) + "=" + str(a-b) + " Starpība") # Pēc analogijas izdarīts tas pats visām nepieciešamām operācijām
print(str(b) + "-" + str(a) + "=" + str(b-a) + " Starpība")
print(str(a) + "*" + str(b) + "=" + str(a*b) + " Reizinājums")
print(str(a) + "/" + str(b) + "=" + str(a/b) + " Dalījums")
print(str(b) + "/" + str(a) + "=" + str(b/a) + " Dalījums")
print(str(a) + "//" + str(b) + "=" + str(a//b) + " Veselais dalījums")
print(str(b) + "//" + str(a) + "=" + str(b//a) + " Veselais dalījums")
print(str(a) + "%" + str(b) + "=" + str(a%b) + " Dalīšanas atlikums")
print(str(b) + "%" + str(a) + "=" + str(b%a) + " Dalīšanas atlikums")
```

Testa piemēri:

1)

```
"Dažas operācijas ar diviem veseliem skaitļiem"
Versija 1.4

Ievadi pirmo veselo skaitli ==>1
Ievadi otro veselo skaitli ==>1
1+1=2 Summa
1-1=0 Starpība
1-1=0 Starpība
1*1=1 Reizinājums
1/1=1.0 Dalījums
1/1=1.0 Dalījums
1//1=1 Veselais dalījums
1//1=1 Veselais dalījums
1%1=0 Dalīšanas atlikums
1%1=0 Dalīšanas atlikums
```

2)

```
"Dažas operācijas ar diviem veseliem skaitļiem"
Versija 1.4

Ievadi pirmo veselo skaitli ==>641641
Ievadi otro veselo skaitli ==>496497
641641+496497=1138138 Summa
641641-496497=145144 Starpība
496497-641641=-145144 Starpība
641641*496497=318572831577 Reizinājums
641641/496497=1.2923361067639885 Dalījums
496497/641641=0.7737925101419642 Dalījums
641641//496497=1 Veselais dalījums
496497//641641=0 Veselais dalījums
641641%496497=145144 Dalīšanas atlikums
496497%641641=496497 Dalīšanas atlikums
```

3)

```
"Dažas operācijas ar diviem veseliem skaitļiem"
Versija 1.4

Ievadi pirmo veselo skaitli ==>100
Ievadi otro veselo skaitli ==>36
100+36=136 Summa
100-36=64 Starpība
36-100=-64 Starpība
100*36=3600 Reizinājums
100/36=2.7777777777777777 Dalījums
36/100=0.36 Dalījums
100//36=2 Veselais dalījums
36//100=0 Veselais dalījums
100%36=28 Dalīšanas atlikums
36%100=36 Dalīšanas atlikums
```

2. uzdevums

Izveidot programmu, kura aprēķina un paziņo divargumentu funkcijas vērtību. Funkcija:

$$f(x, y) = \frac{\log_2(1 + \sqrt[3]{x^2}) + \sin(y)}{|x - y| + 1}$$

Kods:

```
# Programmas nosaukums: Divargumentu funkciju vērtība
```

```
# 2. uzdevums
```

```
# Uzdevuma formulējums: Izveidot programmu, kura aprēķina un paziņo divargumentu funkcijas
vērtību. Funkcija: f(x,y) = (log2(1+cbt(x^2))+sin(y))/(Abs(x-y)+1
```

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

```
# Versija 1.6
```

```
import math # Bibliotēkas math pieslēgšana
```

```
print("\nDivargumentu funkciju vērtība\n\nVersija 1.6\n") # Nosaukums un versija
```

```
x=int(input("Ievadi x vērtību ==>"))
```

```
y=int(input("Ievadi y vērtību ==>"))
```

```
function_value=(math.log2(1+((x**2)**(1/3)))+math.sin(y))/((abs(x-y))+1) # Funkcijas vērtības  
apreķināšana ar funkcijas math.sin un math.log2
```

```
print("Funkcijas vērtība, kad x="+str(x)+" un y="+str(y)+" , ir vienāda ar: "+str(function_value)) # Lai  
savienotu str un int vērtības pārveidosim iegūto funkcijas vērtību par str
```

Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Divargumentu funkciju vērtība  
# 2. uzdevums  
# Uzdevuma formulējums: Izveidot programmu, kura aprēķina un paziņo divargumentu funkcijas vērtību. Funkcija:  $f(x,y) = (\log_2(1+\sqrt[3]{x^2})+\sin(y)) / (|x-y|+1)$   
# Programmas autors: Vladislavs Babajins  
# Versija 1.6  
  
import math # Bibliotēkas math pieslēgšana  
  
print("\nDivargumentu funkciju vērtība\n\nVersija 1.6\n") # Nosaukums un versija  
  
x=int(input("Ievadi x vērtību ==>"))  
y=int(input("Ievadi y vērtību ==>"))  
  
function_value=(math.log2(1+((x**2)**(1/3)))+math.sin(y))/((abs(x-y))+1) # Funkcijas vērtības apreķināšana ar funkcijas math.sin un math.log2  
print("Funkcijas vērtība, kad x="+str(x)+" un y="+str(y)+" , ir vienāda ar: "+str(function_value)) # Lai savienotu str un int vērtības pārveidos
```

Testa piemēri:

1)

```
"Divargumentu funkciju vērtība"  
Versija 1.6  
  
Ievadi x vērtību ==>5  
Ievadi y vērtību ==>0  
Funkcijas vērtība, kad x=5 un y=0, ir vienāda ar: 0.3287219271935167
```

2)

```
"Divargumentu funkciju vērtība"  
Versija 1.6  
  
Ievadi x vērtību ==>0  
Ievadi y vērtību ==>0  
Funkcijas vērtība, kad x=0 un y=0, ir vienāda ar: 0.0
```

3)

```
"Divargumentu funkciju vērtība"  
Versija 1.6  
  
Ievadi x vērtību ==>-15  
Ievadi y vērtību ==>3  
Funkcijas vērtība, kad x=-15 un y=3, ir vienāda ar: 0.15606936843757846
```

3. uzdevums

Izveidot programmu, kura pēc visu trijstūra malu ievades, paziņo tā: perimetru, laukumu, ap to apvilktās riņķa līnijas rādiusu, tajā ievilktais riņķa līniju rādiusu.

Kods:

```
# Programmas nosaukums: Trijstura parametri
```

```
# 3. uzdevums
```

```
# Uzdevuma formulējums: Izveidot programmu, kura pēc visu trijstūra malu ievades, paziņo tā:  
perimetru, laukumu, ap to apvilktās riņķa līnijas rādiusu, tajā ievilktais riņķa līniju rādiusu.
```

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

```
# Versija 2.2
```

```
import math
```

```
print("\nTrijstura parametri\n\nVersija 2.2\n")
```

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

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

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

```
# Trijstura nevienādības pārbaudīšana
```

```
if a >= b+c:
```

```
    print("Tāds trijsturis neēksiste")
```

```
elif b >= a+c:
```

```
    print("Tāds trijsturis neēksiste")
```

```
elif c >= a+b:
```

```
    print("Tāds trijsturis neēksiste")
```

```
else:
```

```
    perimeter=(a+b+c) # Perimetra apreķināšana
```

```
print("Trijstūra perimetrs: "+str(perimeter))
```

```
p=(a+b+c)/2 # Pusperimetra apreķināšana
```

```
area=math.sqrt(p*(p-a)*(p-b)*(p-c)) # Laukuma apreķināšana izmantojot Herona formulu
```

```
print("Laukums: "+str(area))
```

```
R=(a*math.sqrt(3)/3) # Apvilkta riņķa līnijas rādiuss
```

```
print("Apvilkta riņķa līnijas rādiuss: "+str(R))
```

```
r=(a*math.sqrt(3)/6) # Ievilkta riņķa līnijas rādiuss
```

```
print("Ievilkta riņķa līnijas rādiuss: "+str(r))
```

Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Trijstura parametri
# 3. uzdevums
# Uzdevuma formulējums: Izveidot programmu, kura pēc visu trijstūra malu ievades, paziņo tā: perimetru, laukumu, ap to apvilkta riņķa līnijas rādiusu, tajā ievilkta riņķa līniju rādiusu
# Programmas autors: Vladislavs Babanins
# Versija 2.2

import math

print("\nTrijstura parametri\nVersija 2.2\n")

a=float(input("Ievadi 1. malu ==>"))
b=float(input("Ievadi 2. malu ==>"))
c=float(input("Ievadi 3. malu ==>"))

# Trijstura nevienādības pārbaudīšana
if a >= b+c:
    print("Tāds trijsturis neeksistē")
elif b >= a+c:
    print("Tāds trijsturis neeksistē")
elif c >= a+b:
    print("Tāds trijsturis neeksistē")
else:
    perimeter=(a+b+c) # Perimetra apreķināšana
    print("Trijstūra perimetrs: "+str(perimeter))

    p=(a+b+c)/2 # Pusperimetra apreķināšana

    area=math.sqrt(p*(p-a)*(p-b)*(p-c)) # Laukuma apreķināšana izmantojot Herona formulu
    print("Laukums: "+str(area))

    R=(a*math.sqrt(3)/3) # Apvilkta riņķa līnijas rādiuss
    print("Apvilkta riņķa līnijas rādiuss: "+str(R))

    r=(a*math.sqrt(3)/6) # Ievilkta riņķa līnijas rādiuss
    print("Ievilkta riņķa līnijas rādiuss: "+str(r))
```

Testa piemēri:

1)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>3  
Ievadi 2. mālu ==>4  
Ievadi 3. mālu ==>5  
Trijstūra perimetrs: 12.0  
Laukums: 6.0  
Apvilkta riņķa līnijas rādiuss: 1.7320508075688774  
Ievilkta riņķa līnijas rādiuss: 0.8660254037844387
```

2)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>1  
Ievadi 2. mālu ==>30  
Ievadi 3. mālu ==>2  
Tāds trijsturis neēksiste
```

3)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>-3  
Ievadi 2. mālu ==>-4  
Ievadi 3. mālu ==>-5  
Tāds trijsturis neēksiste
```

4)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>4194  
Ievadi 2. mālu ==>4196  
Ievadi 3. mālu ==>165  
Trijstūra perimetrs: 8555.0  
Laukums: 345995.1464954928  
Apvilkta riņķa līnijas rādiuss: 2421.4070289812903  
Ievilkta riņķa līnijas rādiuss: 1210.7035144906451
```

5)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>6  
Ievadi 2. mālu ==>8  
Ievadi 3. mālu ==>10  
Trijstūra perimetrs: 24.0  
Laukums: 24.0  
Apvilkta riņķa līnijas rādiuss: 3.464101615137755  
Ievilkta riņķa līnijas rādiuss: 1.7320508075688774
```

6)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>-3  
Ievadi 2. mālu ==>4  
Ievadi 3. mālu ==>5  
Tāds trijsturis neeksiste
```


7)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>0  
Ievadi 2. mālu ==>0  
Ievadi 3. mālu ==>0  
Tāds trījsturis neēksiste
```

8)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>1  
Ievadi 2. mālu ==>1  
Ievadi 3. mālu ==>1  
Trijstūra perimetrs: 3.0  
Laukums: 0.4330127018922193  
Apvilkta riņķa līnijas rādiuss: 0.5773502691896257  
Ievilkta riņķa līnijas rādiuss: 0.28867513459481287
```

9)

```
"Trījstura paramētri"  
Versija 2.2  
  
Ievadi 1. mālu ==>13.66  
Ievadi 2. mālu ==>10.2  
Ievadi 3. mālu ==>9  
Trijstūra perimetrs: 32.86  
Laukums: 45.89835244091012  
Apvilkta riņķa līnijas rādiuss: 7.886604677130287  
Ievilkta riņķa līnijas rādiuss: 3.9433023385651436
```

4. uzdevums

Izveidot programmu, kura konvertē EUR -> USD un USD -> EUR.

Kods:

```
# Programmas nosaukums: Valūtu konvertācija
# 4. uzdevums
# Uzdevuma formulējums: Izveidot programmu, kura konvertē EUR -> USD un USD -> EUR.
# Programmas autors: Vladislavs Babaņins
# Versija 1.4

# 1 USD = 1.01002 EUR
# 1 EUR = 0.98715628 USD

import math

print("\nValūtu konvertācija\n\nVersija 1.4\n") # Programmas nosaukums

a=int(input("EUR=>USD vai USD=>EUR \nJa EUR=>USD tad 1, ja USD=>EUR tad 2 ==>")) # EUR =>
USD vai USD => EUR izvēle ar 1 vai 2

if a == 1: # Ja tika ievādīts 1 (EUR => USD) tad izpildās komandas:
    EUR=input("EUR=>USD \nEUR vērtība ==>") # EUR vērtību ievādišana
    EUR_to_USD=float(EUR)*0.98715628 # Valūtu konvertācija
    print("Vērtība ir " + str(EUR_to_USD) + " USD") # Ievādišana ekrāna

elif a == 2: # Ja tika ievādīts 2 (USD => EUR) tad izpildās komandas:
    USD=input("USD=>EUR \nUSD vērtība ==>") # USD vērtību ievādišana
    USD_to_EUR=float(USD)*1.01002 # Valūtu konvertācija
    print("Vērtība ir " + str(USD_to_EUR)+ " EUR") # Ievādišana ekrāna
```

else:

```
print("Error") # Ja tika ievadīts ne 1 vai 2, tad ir "Error"
```

Programmas kodu ekrānuņēmums:

```
# Programmas nosaukums: Valūtu konvertācija
# 4. uzdevums
# Uzdevuma formulējums: Izveidot programmu, kura konvertē EUR -> USD un USD -> EUR.
# Programmas autors: Vladislavs Babaiņš
# Versija 1.4

# 1 USD = 1.01002 EUR
# 1 EUR = 0.98715628 USD

import math

print("\nValūtu konvertācija\nVersija 1.4\n") # Programmas nosaukums

a=int(input("EUR=>USD vai USD=>EUR \nJa EUR=>USD tad 1, ja USD=>EUR tad 2 ====>")) # EUR => USD vai USD => EUR izvēle ar 1 vai 2

if a == 1: # Ja tika ievadīts 1 (EUR => USD) tad izpildās komandas:
    EUR=input("EUR=>USD \nEUR vērtība ==>") # EUR vērtību ievādišana
    EUR_to_USD=float(EUR)*0.98715628 # Valūtu konvertācija
    print("Vērtība ir " + str(EUR_to_USD) + " USD") # Ievādišana ekrāna

elif a == 2: # Ja tika ievadīts 2 (USD => EUR) tad izpildās komandas:
    USD=input("USD=>EUR \nUSD vērtība ==>") # USD vērtību ievādišana
    USD_to_EUR=float(USD)*1.01002 # Valūtu konvertācija
    print("Vērtība ir " + str(USD_to_EUR)+" EUR") # Ievādišana ekrāna

else:
    print("Error") # Ja tika ievadīts ne 1 vai 2, tad ir "Error"
```

Testa piemēri:

1)

```
"Valūtu konvertācija"
Versija 1.4

EUR=>USD vai USD=>EUR
Ja EUR=>USD tad 1, ja USD=>EUR tad 2 ====>1
EUR=>USD
EUR vērtība ==>100
Vērtība ir 98.715628 USD
```

2)

```
"Valūtu konvertācija"  
Versija 1.4  
  
EUR=>USD vai USD=>EUR  
Ja EUR=>USD tad 1, ja USD=>EUR tad 2 ==>2  
USD=>EUR  
USD vērtība ==>1000  
Vērtība ir 1010.0199999999999 EUR
```

3)

```
"Valūtu konvertācija"  
Versija 1.4  
  
EUR=>USD vai USD=>EUR  
Ja EUR=>USD tad 1, ja USD=>EUR tad 2 ==>141414  
Error
```

PU1.

Izveidot programmu, kas realizē 4. uzdevumu, izmantojot grafisko saskarni.

Kods:

```
# Programmas nosaukums: Valūtu konvertācija (GUI)  
# 4. uzdevums (PU1)  
# Uzdevuma formulējums: Izveidot programmu, kura konvertē EUR -> USD un USD -> EUR ar GUI.  
# Programmas autors: Vladislavs Babaņins  
# Versija 4.0
```

```
# 1 USD = 1.0129647 EUR
```

```
# 1.00 EUR = 0.98715628 USD
```

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

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

```
root.title("Valūtu konvertācija") # Windows "loga" nosaukums
```

```
# EUR => USD #
```

```
upper_text = Label(root, text="Valūtu konvertācija") # Teksta "Valūtu konvertācija" definēšana
```

```
upper_text.grid(row=0, column=2) # Rāda tekstu 0 rinda, 2 kolonnā
```

```
EUR_label = Label(root, text="EUR") # Teksta "EUR" definēšana
```

```
EUR_label.grid(row=0, column=1) # Rāda tekstu 0 rinda, 1 kolonnā
```

```
USD_label = Label(root, text="USD") # Teksta "EUR" definēšana
```

```
USD_label.grid(row=0, column=3) # Rāda tekstu 0 rinda, 3 kolonnā
```

```
Input_EUR=Entry(root) # Input definēšana (EUR input)
```

```
Input_EUR.insert(0, "EUR vērtība") # 0, "EUR vērtība" - tas kas ir pēc noklusējuma uzreiz ir uzrakstīts  
ailītē
```

```
Input_EUR.get() # Lai "saņemtu" kas bija ierakstīts ailītē
```

```
Input_EUR.grid(row=2, column=1) # Rāda input EUR (ailīte) 2. rinda, 1. kolonnā
```

```
def Click_on_right(): # Definējam komandu EUR -> USD
```

```
    Fuction_EUR_to_USD = Label(root, text=str(float(Input_EUR.get())*0.98715628) + "  
USD").grid(row=3, column=1) # Valūtu konvertācija. Paņemam EUR vērtību no INPUT (kas lietotājs ir  
uzrakstījis). Rezultāta parādīšana 3. rindā un 1. kolonnā ar "USD" tekstu
```

```
# EUR => USD #
```

```
# USD => EUR #
```

```
Input_USD=Entry(root) # Input definēšana (USD input)
```

```
Input_USD.insert(0, "USD vērtība") # 0, "USD vērtība" - tas kas ir pēc noklusējuma uzreiz ir uzrakstīts ailītē
```

```
Input_USD.get() # Lai "saņemtu" kas bija ierakstīts ailītē
```

```
Input_USD.grid(row=2, column=3) # Rāda input (ailīte) EUR 2. rinda, 3. kolonnā
```

```
def Click_on_left(): # Definējam komandu USD -> EUR
```

```
    Fuction_USD_to_EUR = Label(root, text=str(float(Input_USD.get())*1.0129647) + " EUR")  
    ).grid(row=3, column=3) # Valūtu konvertācija. Paņemam USD vērtību no INPUT (kas lietotājs ir uzrakstījis). Rezultāta parādīšana 3. rindā un 3. kolonnā ar "EUR" tekstu
```

```
Kurss = Label(root, text="1.00 EUR = 0.98715628 USD\n1.00 USD = 1.0129647 EUR") # Kursa vērtība uz ekrāna (parasts teksts)
```

```
Kurss.grid(row=4, column=2) # Definēts, kur šis teksts atrodas
```

```
Button_right = Button(root, text="EUR -> USD", command=Click_on_right) # Izmantojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta (EUR -> USD)
```

```
Button_right.grid(row=2, column=2) # Parādam, kur poga tiks attēlota
```

```
Button_left = Button(root, text="USD -> EUR", command=Click_on_left) # Izmantojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta (USD -> EUR)
```

```
Button_left.grid(row=3, column=2) # Parādam, kur poga tiks attēlota
```

USD => EUR

root.mainloop()

Programmas kodu ekrānuņēmums:

```
# 1 USD = 1.0129647 EUR
# 1.00 EUR = 0.98715628 USD

from tkinter import * # Importējam tkinter moduli
root = Tk() # Tkinter (lai izmantotu to komandas)

root.title("Valūtu konvertācija") # Windows "loga" nosaukums

# EUR -> USD #
upper_text = Label(root, text="Valūtu konvertācija") # Teksta "Valūtu konvertācija" definēšana
upper_text.grid(row=0, column=2) # Rāda tekstu 0. rindā, 2. kolonnā

EUR_label = Label(root, text="EUR") # Teksta "EUR" definēšana
EUR_label.grid(row=1, column=1) # Rāda tekstu 1. rindā, 1. kolonnā

USD_label = Label(root, text="USD") # Teksta "USD" definēšana
USD_label.grid(row=1, column=3) # Rāda tekstu 1. rindā, 3. kolonnā

input_EUR_entry(root) # Input definēšana (EUR input)
input_EUR.insert(0, "Eiņa vērtība") # 0, "Eiņa vērtība" - tas kas ir pēc noklusējuma uzreiz ir uzrakstīts ailītē
input_EUR.get() # Lai "sāņemtu" kas bija ierakstīts ailītē
input_EUR.grid(row=2, column=1) # Rāda input (ailīte) 2. rindā, 1. kolonnā

def Click_on_right(): # Definējam komandu EUR -> USD
    fuction_EUR_to_USD = Label(root, text=str(float(input_EUR.get())*0.98715628) + " USD").grid(row=3, column=1) # Valūtu konvertācija. Ņemam EUR vērtību no INPUT (kas lietotājs ir uzrakstījis). Rezultāta parādīšana 3. rindā un 1. kolonnā ar "USD" tekstu

# EUR -> USD #

# USD -> EUR #
input_USD_entry(root) # Input definēšana (USD input)
input_USD.insert(0, "USD vērtība") # 0, "USD vērtība" - tas kas ir pēc noklusējuma uzreiz ir uzrakstīts ailītē
input_USD.get() # Lai "sāņemtu" kas bija ierakstīts ailītē
input_USD.grid(row=2, column=3) # Rāda input (ailīte) EUR 2. rindā, 3. kolonnā

def Click_on_left(): # Definējam komandu USD -> EUR
    fuction_USD_to_EUR = Label(root, text=str(float(input_USD.get())*1.0129647) + " EUR").grid(row=3, column=3) # Valūtu konvertācija. Ņemam USD vērtību no INPUT (kas lietotājs ir uzrakstījis). Rezultāta parādīšana 3. rindā un 3. kolonnā ar "EUR" tekstu

Kursas = Label(root, text="1.00 EUR = 0.98715628 USD un 1.00 USD = 1.0129647 EUR") # Kursu vērtība uz ekrāna (parasts teksts)
Kursas.grid(row=4, column=2) # Definēts, kur šis teksts atrodas

Button_right = Button(root, text="EUR -> USD", command=Click_on_right) # Izņemtojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta (EUR -> USD)
Button_right.grid(row=2, column=2) # Parādam, kur poga tiks attēlota

Button_left = Button(root, text="USD -> EUR", command=Click_on_left) # Izņemtojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta (USD -> EUR)
Button_left.grid(row=2, column=2) # Parādam, kur poga tiks attēlota

# USD -> EUR #

root.mainloop()
```

Testa piemēri:

1)

The screenshot shows a window titled "Valūtu konvertācija". It has three input fields: "EUR" with the value "100", "USD" with the value "100", and a central area with two buttons: "EUR -> USD" and "USD -> EUR". Below the buttons, the text "1.00 EUR = 0.98715628 USD" and "1.00 USD = 1.0129647 EUR" is displayed. The "EUR -> USD" button is highlighted with a red box.

2)

The screenshot shows the same window as before, but with the "USD -> EUR" button highlighted with a red box. The "EUR" input field now contains "0.40" and the "USD" input field contains "0.50". The central text now shows "0.394862512 USD" and "0.50648235 EUR".

3)

EUR	Valūtu konvertācija	USD
10.99	EUR -> USD	10.99
10.8488475172 USD	USD -> EUR	11.132482053 EUR
1.00 EUR = 0.98715628 USD 1.00 USD = 1.0129647 EUR		

GUI:

EUR	Valūtu konvertācija	USD
EUR vērtība	EUR -> USD	USD vērtība
	USD -> EUR	
1.00 EUR = 0.98715628 USD 1.00 USD = 1.0129647 EUR		

EUR	Valūtu konvertācija	USD
1	EUR -> USD	USD vērtība
0.98715628 USD	USD -> EUR	
1.00 EUR = 0.98715628 USD 1.00 USD = 1.0129647 EUR		

EUR	Valūtu konvertācija	USD
0.20	EUR -> USD	1110
0.197431256 USD	USD -> EUR	1124.390817 EUR
1.00 EUR = 0.98715628 USD 1.00 USD = 1.0129647 EUR		

EUR	Valūtu konvertācija	USD
0.255555555555	EUR -> USD	1551511515555
0.2522732715555007 USD	USD -> EUR	15716263969012.223 EUR
1.00 EUR = 0.98715628 USD 1.00 USD = 1.0129647 EUR		