

# 1. uzdevums

Izveidot vienloga lietotni, kas nodrošina divu veselu skaitļu ievadi un pēc pogas piespiešanas paziņo šo skaitļu dalījumu un atlikumu.

## Kods:

```
# Programmas nosaukums: Skaitļu dalījums un atlikums
```

```
# 1. uzdevums (MPR4)
```

```
# Uzdevuma formulējums: Izveidot vienloga lietotni, kas nodrošina divu veselu skaitļu ievadi un pēc  
# pogas piespiešanas paziņo šo skaitļu dalījumu un atlikumu.
```

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

```
# Versija 1.0
```

```
import tkinter
```

```
from tkinter import ttk
```

```
root = tkinter.Tk()
```

```
root.title("Skaitļu dalījums un atlikums")
```

```
root.geometry("220x100")
```

```
def Click_on_button():
```

```
    x=InputLeft.get()
```

```
    y=InputRight.get()
```

```
    z=int(x)//int(y)
```

```
    w=int(x)%int(y)
```

```
    Result.config(text = (str(z) + " atl. " + str(w)))
```

```
Poga=tkinter.Button(root, text="=", command=Click_on_button)
```

```
Poga.pack()
```

```
Poga.place(x=130, y=30, width=25, height=25)
```

```
DalisanasZime=tkinter.Label(root, text=":")
```

```
InputLeft=tkinter.Entry(root)
```

```
InputLeft.pack()
```

```
InputLeft.place(x=50, y=30, width=25, height=25)
```

```
DalisanasZime=tkinter.Label(root, text=":")
```

```
DalisanasZime.pack()
```

```
DalisanasZime.place(x=78, y=30)
```

```
InputRight=tkinter.Entry(root)
```

```
InputRight.pack()
```

```
InputRight.place(x=90, y=30, width=25, height=25)
```

```
Result = ttk.Label(root, text="")
```

```
Result.place(x=165, y=30)
```

```
root.mainloop()
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Skaitļu dalījums un atlikums
# 1. uzdevums (MPR4)
# Uzdevuma formulējums: Izveidot vienloga lietotni, kas nodrošina divu veselu skaitļu ievadi un pēc pogas piespiešanas paziņo šo skaitļu dalījumu un atlikumu.
# Programmas autors: Vladislavs Babaiņš
# Versija 1.0

import tkinter
from tkinter import ttk

root = tkinter.Tk()

root.title("Skaitļu dalījums un atlikums")
root.geometry("220x100")

def Click_on_button():
    x=InputLeft.get()
    y=InputRight.get()

    z=int(x)//int(y)
    w=int(x)%int(y)

    Result.config(text = (str(z) + " atl. " + str(w)))

Poga=tkinter.Button(root, text="=", command=Click_on_button)
Poga.pack()
Poga.place(x=130, y=30, width=25, height=25)

DalisanasZime=tkinter.Label(root, text=":")

InputLeft=tkinter.Entry(root)
InputLeft.pack()
InputLeft.place(x=50, y=30, width=25, height=25)

DalisanasZime=tkinter.Label(root, text=":")
DalisanasZime.pack()
DalisanasZime.place(x=78, y=30)

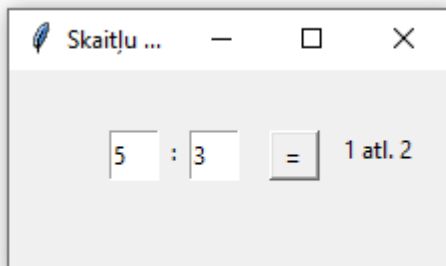
InputRight=tkinter.Entry(root)
InputRight.pack()
InputRight.place(x=90, y=30, width=25, height=25)

Result = ttk.Label(root, text="")
Result.place(x=165, y=30)

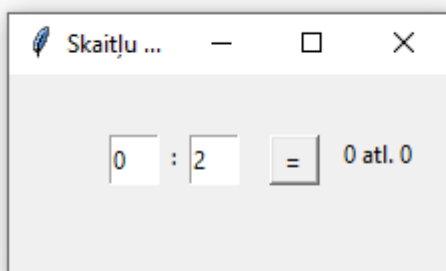
root.mainloop()
```

## Testa piemēri:

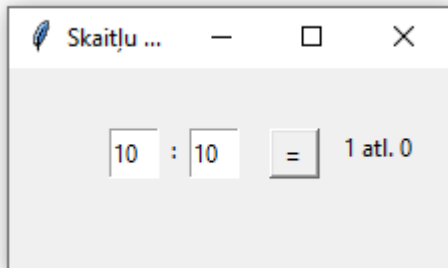
1)



2)



3)



---

## 2. uzdevums

Izveidot programmu, kura aprēķina divargumentu funkcijas vērtību.

$$f(x, y) = \begin{cases} x - y, & \text{ja } x > y \\ x + y, & \text{ja citādi} \end{cases}$$

### Kods:

```
# Programmas nosaukums: Divargumentu funkcijas aprēķināšana
# 2. uzdevums (MPR4)
# Uzdevuma formulējums: Izveidot programmu, kura aprēķina divargumentu funkcijas vērtību.
# Programmas autors: Vladislavs Babaņins
# Versija 1.0

print("\nDivargumentu funkcijas aprēķināšana\n\nVersija 1.0\n")

print("f(x,y) = (x - y, ja x > y) un (x + y, ja citādi)\n")

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

if (x>y):
    print("Divargumentu funkcijas vērtība ir x - y, ja x > y. \nRezultāts: " + str(x-y))
else:
    print("Divargumentu funkcijas vērtība ir x + y, ja x <= y. \nRezultāts: " + str(x+y))
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Divargumentu funkcijas aprēķināšana
# 2. uzdevums (MPR4)
# Uzdevuma formulējums: Izveidot programmu, kura aprēķina divargumentu funkcijas vērtību.
# Programmas autors: Vladislavs Babanins
# Versija 1.0
print("\nDivargumentu funkcijas aprēķināšana\nVersija 1.0\n")

print("f(x,y) = (x - y, ja x > y) un (x + y, ja citādi)\n")

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

if (x>y):
    print("Divargumentu funkcijas vērtība ir x - y, ja x > y. \nRezultāts: " + str(x-y))
else:
    print("Divargumentu funkcijas vērtība ir x + y, ja x <= y. \nRezultāts: " + str(x+y))
```

## Testa piemēri:

1)

```
"Divargumentu funkcijas aprēķināšana"
Versija 1.0

f(x,y) = (x - y, ja x > y) un (x + y, ja citādi)

Ievadi x skaitli ==>10
Ievadi y skaitli ==>3
Divargumentu funkcijas vērtība ir x - y, ja x > y.
Rezultāts: 7.0
```

2)

```
"Divargumentu funkcijas aprēķināšana"
Versija 1.0

f(x,y) = (x - y, ja x > y) un (x + y, ja citādi)

Ievadi x skaitli ==>10
Ievadi y skaitli ==>10
Divargumentu funkcijas vērtība ir x + y, ja x <= y.
Rezultāts: 20.0
```

3)

```
"Divargumentu funkcijas apreķināšana"  
Versija 1.0  
  
f(x,y) = (x - y, ja x > y) un (x + y, ja citādi)  
  
Ievadi x skaitli ==>12.5  
Ievadi y skaitli ==>-13.5  
Divargumentu funkcijas vērtība ir x - y, ja x > y.  
Rezultāts: 26.0
```

4)

```
"Divargumentu funkcijas apreķināšana"  
Versija 1.0  
  
f(x,y) = (x - y, ja x > y) un (x + y, ja citādi)  
  
Ievadi x skaitli ==>10  
Ievadi y skaitli ==>30  
Divargumentu funkcijas vērtība ir x + y, ja x <= y.  
Rezultāts: 40.0
```

5)

```
"Divargumentu funkcijas apreķināšana"  
Versija 1.0  
  
f(x,y) = (x - y, ja x > y) un (x + y, ja citādi)  
  
Ievadi x skaitli ==>0  
Ievadi y skaitli ==>-10  
Divargumentu funkcijas vērtība ir x - y, ja x > y.  
Rezultāts: 10.0
```

---

### 3. uzdevums

Izveidot programmu, kura pieprasa ievadīt divus skaitļus un paziņo, kurš no viņiem ir lielākais.

#### Kods:

```
# Programmas nosaukums: Lielāks skaitlis
```

```
# 3. uzdevums (MPR4)
```

```
# Uzdevuma formulējums: Izveidot programmu, kura pieprasa ievadīt divus skaitļus un paziņo, kurš  
no viņiem ir lielākais.
```

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

```
# Versija 1.0
```

```
print("\nLielāks skaitlis\n\nVersija 1.0\n")
```

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

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

```
if (x>y):
```

```
    print("x > y\nx ir lielāks par y")
```

```
if (x<y):
```

```
    print("x < y\ny ir lielāks par x")
```

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

```
    print("x==y\ny Nav lielāka skaitļa")
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Lielāks skaitlis
# 3. uzdevums (MPR4)
# Uzdevuma formulējums: Izveidot programmu, kura pieprasa ievadīt divus skaitļus un paziņo, kurš no viņiem ir lielāks
# Programmas autors: Vladislavs Babaņins
# Versija 1.0

print("\nLielāks skaitlis\n\nVersija 1.0\n")

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

if (x>y):
    print("x > y\nx ir lielāks par y")
if (x<y):
    print("x < y\ny ir lielāks par x")
if (x==y):
    print("x==y\nNav lielāka skaitļa")
```

## Testa piemēri:

1)

```
"Lielāks skaitlis"
Versija 1.0

Ievadi x skaitli ==>1
Ievadi y skaitli ==>2
x < y
y ir lielāks par x
```

2)

```
"Lielāks skaitlis"
Versija 1.0

Ievadi x skaitli ==>2
Ievadi y skaitli ==>1
x > y
x ir lielāks par y
```



3)

```
"Lielāks skaitlis"  
Versija 1.0  
  
Ievadi x skaitli ==>0  
Ievadi y skaitli ==>0  
x==y  
y Nav lielāka skaitļa
```

4)

```
"Lielāks skaitlis"  
Versija 1.0  
  
Ievadi x skaitli ==>-12.3  
Ievadi y skaitli ==>-21313  
x > y  
x ir lielāks par y
```

5)

```
"Lielāks skaitlis"  
Versija 1.0  
  
Ievadi x skaitli ==>-1000000  
Ievadi y skaitli ==>1000  
x < y  
y ir lielāks par x
```

6)

```
"Lielāks skaitlis"  
Versija 1.0  
  
Ievadi x skaitli ==>88  
Ievadi y skaitli ==>99  
x < y  
y ir lielāks par x
```

---

## 4. uzdevums

Izveidot programmu, kura pieprasa ievadīt trīs skaitļus, sakārto un nodrukā tos dilstoša secībā.

### Kods:

```
# Programmas nosaukums: Sakartot trīs skaitļus dilstoša secība
```

```
# 4. uzdevums (MPR4)
```

```
# Uzdevuma formulējums: Izveidot programmu, kura pieprasa ievadīt trīs skaitļus, sakārto un  
nodrukā tos dilstoša secībā.
```

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

```
# Versija 1.0
```

```
print("\nSakartot trīs skaitļus dilstoša secība\n\nVersija 1.0\n")
```

```
a = float(input("Ievadiet 1.skaitli ==> "))
```

```
b = float(input("Ievadiet 2.skaitli ==> "))
```

```
c = float(input("Ievadiet 3.skaitli ==> "))
```

```
if a < b :
```

```
    p = a
```

```
    a = b
```

```
    b = p
```

```
if a < c :
```

```
    p = c
```

```
    c = a
```

```
    a = p
```

```
if b < c :
```

```
    p = b
```

```
    b = c
```

```
    c = p
```

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

### Programmas kodu ekrānuzņēmums:

```
print("\nSakartot trīs skaitļus dilstoša secība\n\nVersija 1.0\n")

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

if a < b :
    p = a
    a = b
    b = p

if a < c :
    p = c
    c = a
    a = p

if b < c :
    p = b
    b = c
    c = p

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

## Testa piemēri:

1)

```
"Sakartot trīs skaitļus dilstoša secība"  
Versija 1.0  
  
Ievadiet 1.skaitli ==> 30  
Ievadiet 2.skaitli ==> 10  
Ievadiet 3.skaitli ==> 20  
30.0 20.0 10.0
```

2)

```
"Sakartot trīs skaitļus dilstoša secība"  
Versija 1.0  
  
Ievadiet 1.skaitli ==> 0  
Ievadiet 2.skaitli ==> 1  
Ievadiet 3.skaitli ==> -1  
1.0 0.0 -1.0
```

3)

```
"Sakartot trīs skaitļus dilstoša secība"  
Versija 1.0  
  
Ievadiet 1.skaitli ==> 0  
Ievadiet 2.skaitli ==> 0  
Ievadiet 3.skaitli ==> 0  
0.0 0.0 0.0
```

4)

```
"Sakartot trīs skaitļus dilstoša secība"  
Versija 1.0  
  
Ievadiet 1.skaitli ==> 1.33  
Ievadiet 2.skaitli ==> 1.3  
Ievadiet 3.skaitli ==> 1  
1.33 1.3 1.0
```

5)

```
"Sakārtot trīs skaitļus dilstoša secība"
Versija 1.0

Ievadiet 1.skaitli ==> -10
Ievadiet 2.skaitli ==> 0
Ievadiet 3.skaitli ==> 124.124124124124124
124.12412412412412 0.0 -10.0
```

---

## 5. uzdevums

Izveidot programmu, kura pieprasa ievadīt četrus skaitļus, sakārto un nodrūka tos dilstoša secībā, izmantojot ne vairāk kā sešas salīdzināšanas darbības.

### Kods:

```
# Programmas nosaukums: Sakārtot četrus skaitļus dilstoša secība
```

```
# 5. uzdevums (MPR4)
```

```
# Uzdevuma formulējums: Izveidot programmu, kura pieprasa ievadīt četrus skaitļus, sakārto un
nodrukā tos dilstoša secībā, izmantojot ne vairāk kā sešas salīdzināšanas darbības.
```

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

```
# Versija 1.0
```

```
a = float(input("Ievadiet 1.skaitli ==> "))
```

```
b = float(input("Ievadiet 2.skaitli ==> "))
```

```
c = float(input("Ievadiet 3.skaitli ==> "))
```

```
d = float(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))
```

## Programmas kodu ekrānuzņēmums:

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

## Testa piemēri:

1)

```
Ievadiet 1.skaitli ==> 13
Ievadiet 2.skaitli ==> 2
Ievadiet 3.skaitli ==> 1
Ievadiet 4.skaitli ==> 626
626.0 13.0 2.0 1.0
```

2)

```
Ievadiet 1.skaitli ==> 111
Ievadiet 2.skaitli ==> 0
Ievadiet 3.skaitli ==> -23
Ievadiet 4.skaitli ==> -41124
111.0 0.0 -23.0 -41124.0
```

3)

```
Ievadiet 1.skaitli ==> -1.3
Ievadiet 2.skaitli ==> 11111
Ievadiet 3.skaitli ==> -124124124
Ievadiet 4.skaitli ==> 0
11111.0 0.0 -1.3 -124124124.0
```

## 6. uzdevums

Izveidot programmu, kura pieprasa ievadīt četrus skaitļus, sakārto un nodrukā tos dilstoša secībā, izmantojot ne vairāk kā piecas salīdzināšanas darbības.

### Kods:

```
# Programmas nosaukums: Sakārtot četrus skaitļus dilstoša secībā
```

```
# 6. uzdevums (MPR4)
```

```
# Uzdevuma formulējums: Izveidot programmu, kura pieprasa ievadīt četrus skaitļus, sakārto un
nodrukā tos dilstoša secībā, izmantojot ne vairāk kā piecas salīdzināšanas darbības.
```

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

```
# Versija 1.0
```

```
a = float(input("Ievadiet 1.skaitli ==> "))
```

```
b = float(input("Ievadiet 2.skaitli ==> "))
```

```
c = float(input("Ievadiet 3.skaitli ==> "))
```

```
d = float(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))
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Sakārtot četrus skaitļus dilstoša secībā
# 6. uzdevums (MPR4)
# Uzdevuma formulējums: Izveidot programmu, kura pieprasa ievadīt četrus skaitļus, sakārto un nodrukā tos dilstoša secībā, izmantojot ne vairāk kā pi
# Programmas autors: Vladislavs Babaiņš
# Versija 1.0

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

## Testa piemēri:

1)

```
Ievadiet 1.skaitli ==> 100
Ievadiet 2.skaitli ==> 1
Ievadiet 3.skaitli ==> 20
Ievadiet 4.skaitli ==> -3
100.0 20.0 1.0 -3.0
```

2)

```
Ievadiet 1.skaitli ==> 1
Ievadiet 2.skaitli ==> 20
Ievadiet 3.skaitli ==> -34
Ievadiet 4.skaitli ==> 2
20.0 2.0 1.0 -34.0
```

3)

```
Ievadiet 1.skaitli ==> 0
Ievadiet 2.skaitli ==> -14214.3512
Ievadiet 3.skaitli ==> -12.1231321
Ievadiet 4.skaitli ==> 2
2.0 0.0 -12.1231321 -14214.3512
```

## 7. uzdevums

Sastādīt programmu, kas pieprasa ievadīt temperatūras skaitlisko vērtību un mērvienību, un veic konvertāciju uz citām mērvienībām.

### Kods:

```
# Programmas nosaukums: Temperatūras konvertācija
```

```
# 7. uzdevums (MPR4)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas pieprasa ievadīt temperatūras skaitlisko vērtību
un mērvienību, un veic konvertāciju uz citām mērvienībām.
```

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

```
# Versija 1.4
```

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

```
a=int(input("Izvēlies kādu mērvienību grībi konvertēt citās!\n°C, °F, K\n\nJa °C tad 1, ja °F tad 2, ja K tad 3 ==> ")) # EUR => USD vai USD => EUR izvēle ar 1 vai 2
```

```
if a == 1:
```

```
    C=input("Ievādi temperatūras skaitlisko vērtību!\n°C = ")
```

```
    if float(C)>=-273.15:
```

```
        C_to_F = 1.8*float(C) + 32
```

```
        C_to_K = float(C) + 273.15
```

```
        print(str(C) + " °C = " + str(C_to_F) + " °F") # Ievadišana ekrāna
```

```
        print(str(C) + " °C = " + str(C_to_K) + " K") # Ievadišana ekrāna
```

```
    else:
```

```
        print("Error. Vērtība ir mazāka nekā absolūtā nulle.")
```

```
if a == 2:
```

```
    F=input("\n°F = ")
```

```
    if float(F)>=-459.67:
```

```
        F_to_C=((float(F)-32)/1.8) # Temperatūru konvertācija
```

```
        F_to_K=(float(F)+459.67)/1.8 # Temperatūru konvertācija
```

```
        print(str(F) + " °F = " + str(F_to_C) + " °C") # Ievadišana ekrāna
```

```
        print(str(F) + " °F = " + str(F_to_K) + " K") # Ievadišana ekrāna
```

```
    else:
```

```
        print("Error. Vērtība ir mazāka nekā absolūtā nulle.")
```

if a == 3: # Ja tika ievādīts 3 tad izpildās komandas:

```
K=input("K = ") # K vērtību ievādišana
```

if float(K)>=0: # Absolūtas nulles pārbaudīšana

```
K_to_C=float(K)-273.15 # Temperatūru konvertācija
```

```
K_to_F=1.8*(float(K))-459.67 # Temperatūru konvertācija
```

```
print(str(K) + " K = " + str(K_to_C) + " °C") # Ievadišana ekrāna
```

```
print(str(K) + " K = " + str(K_to_F) + " °F") # Ievadišana ekrāna
```

else:

```
print("Error. Vērtība ir mazāka nekā absolūtā nulle.")
```

if a!=1 and a!=2 and a!=3:

```
print("Error") # Ja tika ievādīts ne 1, ne 2, ne 3, tad ir "Error"
```

## Programmas kodu ekrānuzņēmums:

```
# Programmas nosaukums: Temperatūras konvertācija
# 7. uzdevums (MPR4)
# Uzdevuma formulējums: Sastādīt programmu, kas pieprasa ievadīt temperatūras skaitlisko vērtību un mērvienību, un veic konvertāciju uz citām mērvienībām.
# Programmas autors: Vladislavs Babanins
# Versija 1.4

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

a=int(input("Izvēlies kādu mērvienību gribi konvertēt citās!\n°C, °F, K\n\nJa °C tad 1, ja °F tad 2, ja K tad 3 ----> ")) # EUR -> USD vai USD -> EUR izvēle ar 1 vai 2
if a == 1:
    C=input("Ievadi temperatūras skaitlisko vērtību!\n°C = ")

    if float(C)>=-273.15:

        C_to_F = 1.8*float(C) + 32
        C_to_K = float(C) + 273.15
        print(str(C) + " °C = " + str(C_to_F) + " °F") # Ievadišana ekrāna
        print(str(C) + " °C = " + str(C_to_K) + " K") # Ievadišana ekrāna

    else:
        print("Error. Vērtība ir mazāka nekā absolūtā nulle.")

if a == 2:
    F=input("°F = ")

    if float(F)>=-459.67:

        F_to_C=((float(F)-32)/1.8) # Temperatūru konvertācija
        F_to_K=(float(F)+459.67)/1.8 # Temperatūru konvertācija
        print(str(F) + " °F = " + str(F_to_C) + " °C") # Ievadišana ekrāna
        print(str(F) + " °F = " + str(F_to_K) + " K") # Ievadišana ekrāna

    else:
        print("Error. Vērtība ir mazāka nekā absolūtā nulle.")

if a == 3: # Ja tika ievādīts 3 tad izpildās komandas:
    K=input("K = ") # K vērtību ievādišana

    if float(K)>=0: # Absolūtas nulles pārbaudīšana

        K_to_C=float(K)-273.15 # Temperatūru konvertācija
        K_to_F=1.8*(float(K))-459.67 # Temperatūru konvertācija
        print(str(K) + " K = " + str(K_to_C) + " °C") # Ievadišana ekrāna
        print(str(K) + " K = " + str(K_to_F) + " °F") # Ievadišana ekrāna

    else:
        print("Error. Vērtība ir mazāka nekā absolūtā nulle.")

if a!=1 and a!=2 and a!=3:
    print("Error") # Ja tika ievādīts ne 1, ne 2, ne 3, tad ir "Error"
```

## Testa piemēri:

1)

```
"Temperatūras konvertācija"  
Versija 1.4  
  
Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K  
  
Ja °C tad 1, ja °F tad 2, ja K tad 3 ==> 1  
Ievādi temperatūras skaitlisko vērtību!  
°C = 100  
100 °C = 212.0 °F  
100 °C = 373.15 K
```

2)

```
"Temperatūras konvertācija"  
Versija 1.4  
  
Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K  
  
Ja °C tad 1, ja °F tad 2, ja K tad 3 ==> 2  
°F = 0  
0 °F = -17.77777777777778 °C  
0 °F = 255.37222222222223 K
```

3)

"Temperatūras konvertācija"

Versija 1.4

Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K

Ja °C tad 1, ja °F tad 2, ja K tad 3 ==> 3

K = 0

0 K = -273.15 °C

0 K = -459.67 °F

4)

"Temperatūras konvertācija"

Versija 1.4

Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K

Ja °C tad 1, ja °F tad 2, ja K tad 3 ==> 3

K = -100

Error. Vērtība ir mazāka nekā absolūtā nulle.

5)

"Temperatūras konvertācija"

Versija 1.4

Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K

Ja °C tad 1, ja °F tad 2, ja K tad 3 ==> 1

Ievādi temperatūras skaitlisko vērtību!

°C = -20000

Error. Vērtība ir mazāka nekā absolūtā nulle.

6)

```
"Temperatūras konvertācija"  
Versija 1.4  
  
Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K  
  
Ja °C tad 1, ja °F tad 2, ja K tad 3 ====> 2  
°F = -3000000  
Error. Vērtība ir mazāka nekā absolūtā nulle.
```

7)

```
"Temperatūras konvertācija"  
Versija 1.4  
  
Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K  
  
Ja °C tad 1, ja °F tad 2, ja K tad 3 ====> 5  
Error
```

8)

```
"Temperatūras konvertācija"  
Versija 1.4  
  
Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K  
  
Ja °C tad 1, ja °F tad 2, ja K tad 3 ====> -241  
Error
```

9)

```
"Temperatūras konvertācija"  
Versija 1.4  
  
Izvēlies kādu mērvienību grībi konvertēt citās!  
°C, °F, K  
  
Ja °C tad 1, ja °F tad 2, ja K tad 3 ====> 3  
K = 3000000000000000000000000  
3000000000000000000000000 K = 3e+24 °C  
3000000000000000000000000 K = 5.4e+24 °F
```

10)

```
"Temperatūras konvertācija"
Versija 1.4

Izvēlies kādu mērvienību grībi konvertēt citās!
°C, °F, K

Ja °C tad 1, ja °F tad 2, ja K tad 3 ====> 1
Ievādi temperatūras skaitlisko vērtību!
°C = -273.15
-273.15 °C = -459.66999999999996 °F
-273.15 °C = 0.0 K
```

11)

```
"Temperatūras konvertācija"
Versija 1.4

Izvēlies kādu mērvienību grībi konvertēt citās!
°C, °F, K

Ja °C tad 1, ja °F tad 2, ja K tad 3 ====> 2
°F = -388
-388 °F = -233.33333333333331 °C
-388 °F = 39.816666666666668 K
```

## PU 7 uzd.

Sastādīt programmu, kas pieprasa ievadīt temperatūras skaitlisko vērtību un mērvienību, un veic konvertāciju uz citām mērvienībām ar GUI.

### Kods:

```
# Programmas nosaukums: Temperatūras konvertācija ar GUI
```

```
# PU 7. uzdevums (MPR4)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas pieprasa ievadīt temperatūras skaitlisko vērtību un mērvienību, un veic konvertāciju uz citām mērvienībām ar GUI.
```

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

```
# Versija 6.0
```

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



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

```
root.geometry("430x165") # Loga izmēra definēšana
```

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

```
# Labels
```

```
Temperaturu_konvertacija_label = Label(root, text="Temperatūru konvertācija") # Teksta  
"Temperatūru konvertācija" definēšana
```

```
Temperaturu_konvertacija_label.place(x=147, y=0) # Rāda tekstu uz attiecīgām koordinātām
```

```
C_label = Label(root, text="°C") # Teksta "°C" definēšana
```

```
C_label.place(x=60, y=25) # Rāda tekstu uz attiecīgām koordinātām (atrodas virs pogam)
```

```
F_label = Label(root, text="°F") # Teksta "°F" definēšana
```

```
F_label.place(x=200, y=25) # Rāda tekstu uz attiecīgām koordinātām (atrodas virs pogam)
```

```
K_label = Label(root, text="K") # Teksta "K" definēšana
```

```
K_label.place(x=355, y=25) # Rāda tekstu uz attiecīgām koordinātām (atrodas virs pogam)
```

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

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

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

```
Input_C.place(x=10,y=50) # Rāda input (ailīte) EUR 2. rinda, 3. kolonnā
```

```
Input_F=Entry(root) # Input definēšana (Fārenheita)
```

```
Input_F.insert(0, "°F vērtība") # 0, "°F vērtība" - tas kas ir pēc noklusējuma uzreiz ir uzrakstīts ailītē
```

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

```
Input_F.place(x=155, y=50) # Rāda input Fārenheita(ailīte)
```

```
Input_K=Entry(root) # Input definēšana (Kelvin input)
```

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

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

```
Input_K.place(x=300,y=50) # Rāda input Kelvinos (ailīte)
```

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

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

```
l1.place(x=10, y=110)
```

```
l2 = Label(root, text="")
```

```
l2.place(x=10, y=130)
```

```
l3 = Label(root, text="")
```

```
l3.place(x=165, y=110)
```

```
l4 = Label(root, text="")
```

```
l4.place(x=165, y=130)
```

```
l5 = Label(root, text="")
```

```
l5.place(x=300, y=110)
```

```
l6 = Label(root, text="")
```

```
l6.place(x=300, y=130)
```

```
l7 = Label(root, text="")
```

```
l7.place(x=100, y=125)
```

```
def Click_on_C(): # Definējam komandu Click_on_C
```

```
l1.config(text = "") # Iepriekšējo rezultātu notīrīšana
```

```
l2.config(text = "")
```

```
l7.config(text = "")
```

```
if float(Input_C.get())>=-273.15: # Ja vērtība ir lielāka vai vienāda ar absolūto nulli, tad konvertēsim to
```

```
    a_float = float(Input_C.get())*1.8+32
```

```
    a_round = "{:.2f}".format(a_float) # Lai rezultāts būtu 2 cipāru pēc komata
```

```
    a = str(Input_C.get()) + "°C = " + str(float(a_round)) + " °F"
```

```
    l1.config(text=str(a)) # Rezultātu parādīšana
```

```
    b_float = float(Input_C.get())+273.15
```

```
    b_round = "{:.2f}".format(b_float) # Lai rezultāts būtu 2 cipāru pēc komata
```

```
    b = str(Input_C.get()) + "°C = " + str(float(b_round)) + " K"
```

```
    l2.config(text=str(b)) # Izmainīt tekstu ""
```

```
    Input_C.delete(0, "end") # Nodzēst ierakstīto
```

```
else: # Ja vērtība ir lielāka vai vienāda ar absolūto nulli, tad konvertēsim to
```

```
    l7.config(text = "") # Visu iepriekšējo rezultātu notīrīšana
```

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

```
    l2.config(text = "")
```

```
l3.config(text = "")
l4.config(text = "")
l5.config(text = "")
l6.config(text = "")

l7.config(text="Error. Vērtība ir mazākā nekā absolūtā nulle")
Input_C.delete(0, "end") # Nodzēst ierakstīto
```

```
def Click_on_F(): # Definējam komandu Click_on_F
```

```
l3.config(text = "")
l4.config(text = "")
l7.config(text = "")
```

```
if float(Input_F.get())>=-459.67:
```

```
c_float = ((float(Input_F.get())-32)/1.8)
c_round = "{:.2f}".format(c_float) # Lai rezultāts būtu 2 cipāru pēc komata
```

```
c = str(Input_F.get()) + "°F = " + str(float(c_round)) + " °C"
l3.config(text=str(c)) # Teksta "" izmaiņošana
```

```
d_float = ((float(Input_F.get())+459.67)/1.8)
d_round = "{:.2f}".format(d_float) # Lai rezultāts būtu 2 cipāru pēc komata
```

```
d = str(Input_F.get()) + "°F = " + str(float(d_round)) + " K"
l4.config(text=str(d)) # Teksta "" izmaiņošana
```

```
Input_F.delete(0, "end") # Nodzēst ierakstīto
```

else:

```
l7.config(text = "") # Visu iepriekšējo rezultātu notīrīšana
```

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

```
l2.config(text = "")
```

```
l3.config(text = "")
```

```
l4.config(text = "")
```

```
l5.config(text = "")
```

```
l6.config(text = "")
```

```
l7.config(text="Error. Vērtība ir mazākā nekā absolūtā nulle")
```

```
Input_F.delete(0, "end") # Nodzēst ierakstīto
```

```
def Click_on_K(): # Definējam komandu Click_on_K
```

```
l5.config(text = "")
```

```
l6.config(text = "")
```

```
l7.config(text = "")
```

```
if float(Input_K.get())>=0:
```

```
    e_float = float(Input_K.get())-273.15
```

```
    e_round = "{:.2f}".format(e_float) # Lai rezultāts būtu 2 cipāru pēc komata
```

```
    e = str(Input_K.get()) + "K = " + str(float(e_round)) + " °C"
```

```
    l5.config(text=str(e)) # Teksta "" izmainīšana
```

```
f_float = float(Input_K.get())
f_float2 = (((f_float)*1.8)-459.67)
f_round = "{:.2f}".format(f_float2) # Lai rezultāts būtu 2 cipāru pēc komata
```

```
f = str(Input_K.get()) + "K = " + str(float(f_round)) + " °F"
l6.config(text=str(f)) # Teksta "" izmaiņošana
Input_K.delete(0, "end") # Nodzēst ierakstīto
```

```
else: # Ja vērtība ir mazāka tad ERROR.
```

```
l7.config(text = "") # Visu iepriekšējo rezultātu notīrīšana
```

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

```
l2.config(text = "")
```

```
l3.config(text = "")
```

```
l4.config(text = "")
```

```
l5.config(text = "")
```

```
l6.config(text = "")
```

```
l7.config(text="Error. Vērtība ir mazākā nekā absolūtā nulle")
```

```
Input_K.delete(0, "end") # Nodzēst ierakstīto
```

```
# Pogu definēšana
```

```
C_button = Button(root, text="Konvertēt °C", command=Click_on_C) # Izmantojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta (Click_on_C)
```

```
C_button.place(x=20, y=75, width=100) # Parādam, kur poga tiks attēlota
```

```
F_button = Button(root, text="Konvertēt °F", command=Click_on_F) # Izmantojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta (Click_on_F)
```

```
F_button.place(x=165, y=75, width=100) # Parādam, kur poga tiks attēlota
```

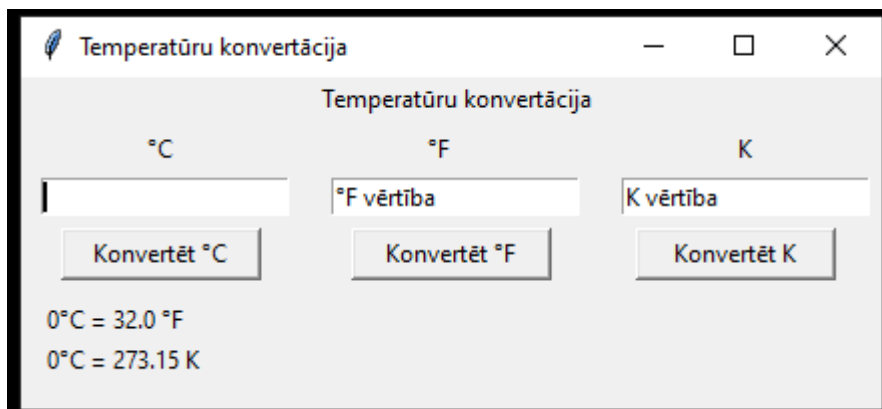
```
K_button = Button(root, text="Konvertēt K", command=Click_on_K) # Izmantojam definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta (Click_on_K)
```

```
K_button.place(x=307, y=75, width=100) # Parādam, kur poga tiks attēlota
```

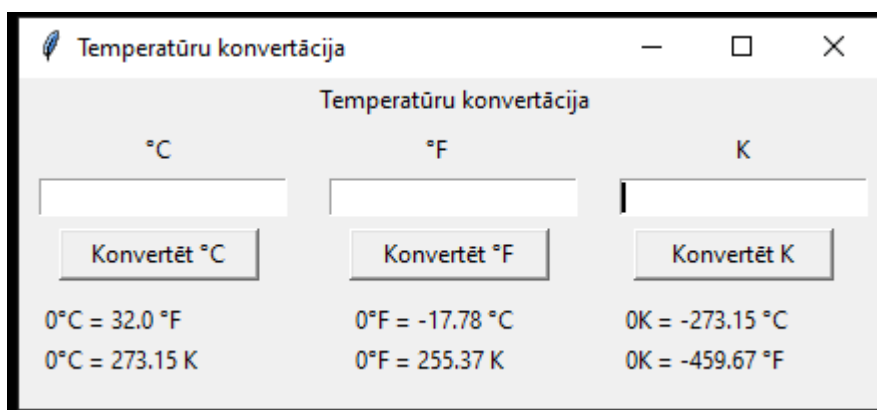
```
root.mainloop() # Obligāta rindiņa, lai logs butu redzāms visu laiku
```

## Testa piemēri:

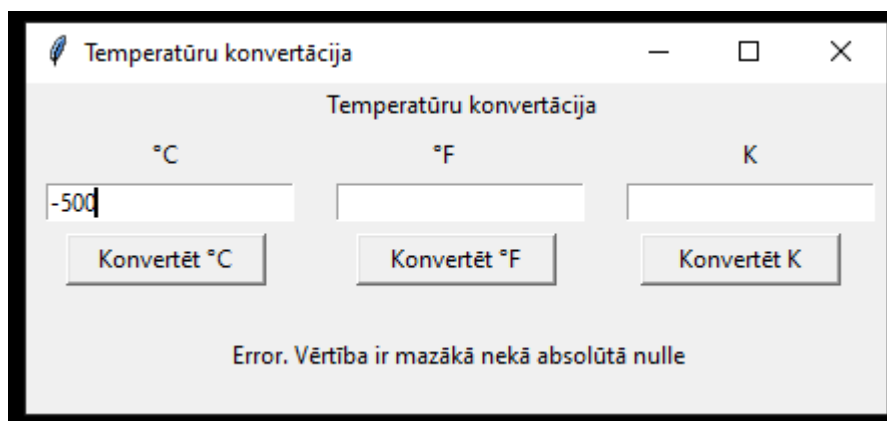
1)



2)

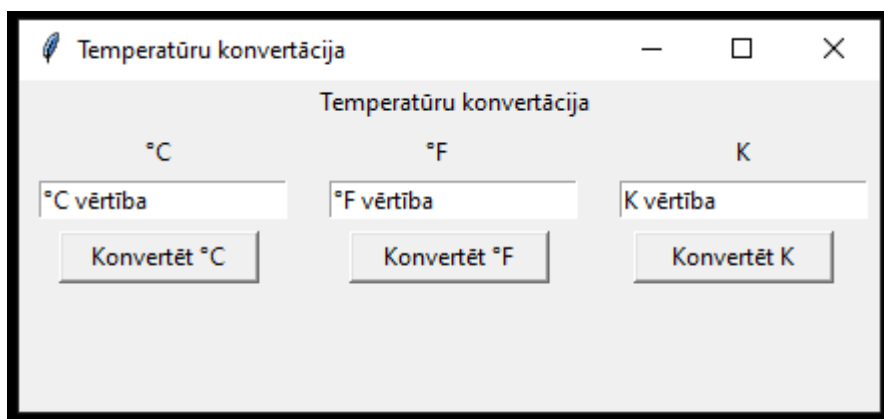


3)

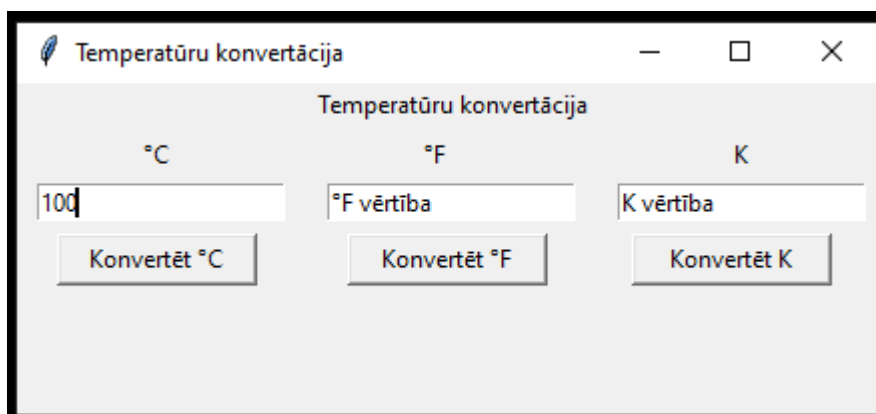


The screenshot shows a window titled "Temperatūru konvertācija" (Temperature Conversion). It has three input fields for temperature units: °C, °F, and K. The °C field contains the value "-500". Below each input field is a button labeled "Konvertēt" (Convert) followed by the unit name. At the bottom of the window, there is an error message: "Error. Vērtība ir mazākā nekā absolūtā nulle" (Error. Value is less than absolute zero).

GUI:



The screenshot shows the same "Temperatūru konvertācija" window. The input fields now contain placeholder text: "°C vērtība" (°C value), "°F vērtība" (°F value), and "K vērtība" (K value). The "Konvertēt" buttons remain below each field.



The screenshot shows the "Temperatūru konvertācija" window with the °C input field containing the value "100". The other fields still have placeholder text, and the "Konvertēt" buttons are visible below them.



Temperatūru konvertācija

Temperatūru konvertācija

°C	°F	K
<input type="text"/>	<input type="text"/>	<input type="text"/>
Konvertēt °C	Konvertēt °F	Konvertēt K
100°C = 212.0 °F	100°F = 37.78 °C	100K = -173.15 °C
100°C = 373.15 K	100°F = 310.93 K	100K = -279.67 °F

Temperatūru konvertācija

Temperatūru konvertācija

°C	°F	K
<input type="text"/>	<input type="text"/>	<input type="text"/>
Konvertēt °C	Konvertēt °F	Konvertēt K
0°C = 32.0 °F	0°F = -17.78 °C	0K = -273.15 °C
0°C = 273.15 K	0°F = 255.37 K	0K = -459.67 °F

Temperatūru konvertācija

Temperatūru konvertācija

°C	°F	K
<input type="text"/>	<input type="text"/>	<input type="text"/>
Konvertēt °C	Konvertēt °F	Konvertēt K
199°C = 390.2 °F	23°F = -5.0 °C	14214K = 13940.85 °C
199°C = 472.15 K	23°F = 268.15 K	14214K = 25125.53 °F

Temperatūru konvertācija

Temperatūru konvertācija

°C	°F	K
<input type="text"/>	<input type="text"/>	<input type="text" value="-300"/>
Konvertēt °C	Konvertēt °F	Konvertēt K

Temperatūru konvertācija

Temperatūru konvertācija

°C	°F	K
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Konvertēt °C"/>	<input type="button" value="Konvertēt °F"/>	<input type="button" value="Konvertēt K"/>

Error. Vērtība ir mazākā nekā absolūtā nulle

Temperatūru konvertācija

Temperatūru konvertācija

°C	°F	K
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Konvertēt °C"/>	<input type="button" value="Konvertēt °F"/>	<input type="button" value="Konvertēt K"/>

151515°F = 84157.22 °C  
151515°F = 84430.37 K