

## 1MPR04\_Simona\_Blinova sb24037

Visos uzdevumos lietoju savu moduli ar funkcijām. (programmu mapē *funkcijas.py*)

### 1.uzdevums

Programma ar funkcijām masīva izveidei, aizpildīšanai ar datiem, izvadei un mazāka un lielāka atrašanai.

#### Kods:

Funkcijas no *funkcijas.py*:

```
# Funkcija pārbaudei, lai skaitlis būtu reāls skaitlis
def reals_skaitlis(a):
    # meģinājumu skaitītājs (3 meģinājumi pareizi ievadīt)
    meģinajumi = 1
    while meģinajumi <= 3:
        try:
            a = float(a)
            return float(a)
        except:
            if meģinajumi < 3:
                meģinajumi += 1
                a = input('Ievadiet reālo skaitli vēlreiz --> ')
            else:
                print('Programma beidz darbību!')
                exit()
```

1MPR04\_1\_Simona\_Blinova.py:

```
import numpy
import funkcijas
```

# Masīva izveides funkcija

```
def masiva_izveide(n):
    return numpy.arange(n)
```

# Datu ievades masīvā funkcija

```
def datu_ievade_masiva(a):
    garums = len(a)
    for i in range(garums):
        vertiba = input('Ievadiet vērtību --> ')
        # pārbaude lai reāls, lai varētu atrast min un max
        vertiba = funkcijas.reals_skaitlis(vertiba)
        a[i] = vertiba
    return a
```

```

# Datu izvades funkcija
def datu_izvade(a):
    print(a)

# Masīva mazākas vērtības atrašanas funkcija
def mazaka_vertiba(a):
    garums = len(a)
    for i in range(garums):
        if i == 0:
            mazakais = a[i]
        else:
            if mazakais > a[i]:
                mazakais = a[i]
    return mazakais

# Masīva lielākas vērtības atrašanas funkcija
def lielaka_vertiba(a):
    garums = len(a)
    for i in range(garums):
        if i == 0:
            lielakais = a[i]
        else:
            if lielakais < a[i]:
                lielakais = a[i]
    return lielakais

n = input('Ievadiet elementu skaitu masīvā --> ')
n = funkcijas.reals_skaitlis(n)

masivs = masiva_izveide(n)
masivs = datu_ievade_masiva(masivs)
datu_izvade(masivs)
print(f'Mazāka vērtība masīva: {mazaka_vertiba(masivs)}')
print(f'Lielāka vērtība masīva: {lielaka_vertiba(masivs)}')

```

Testa piemērs(1)

```

Ievadiet elementu skaitu masīvā --> 3
Ievadiet vērtību --> 1
Ievadiet vērtību --> 2
Ievadiet vērtību --> 3
[1. 2. 3.]
1.0
3.0

```

Testa piemērs(2)

```

Ievadiet elementu skaitu masīvā --> 6
Ievadiet vērtību --> 1.23
Ievadiet vērtību --> 4.51
Ievadiet vērtību --> 64.3
Ievadiet vērtību --> 98.112
Ievadiet vērtību --> 7.3
Ievadiet vērtību --> 0.98
[ 1.23  4.51  64.3  98.112  7.3   0.98 ]
0.98
98.112

```

Testa piemērs(3)

```

Ievadiet elementu skaitu masīvā --> 1
Ievadiet vērtību --> 54.97
[54.97]
54.97
54.97

```

## **2.uzdevums**

Programma ar funkcijām masīva izveidei, aizpildīšanai ar datiem, izvadei un mazāka un lielāka atrašanai.

### **Kods:**

Funkcijas no *funkcijas.py*:

```

# Funkcija pārbaudei, lai skaitlis būtu naturāls skaitlis
def naturals_skaitlis(a):
    # meģinājumu skaita skaitītājs (3 meģinājumi ierakstam)
    meģinajumi = 1
    while meģinajumi <= 3:
        try:
            a = int(a)
            if a > 0:
                return int(a)
            else:
                raise Exception
        except:
            if meģinajumi < 3:
                meģinajumi += 1
                a = input('Ievadiet naturālo skaitli vēlreiz --> ')
            else:
                print('Programma beidz darbību!')
                exit()

```

*1MPR04\_2\_Simona\_Blinova.py*:

```

import funkcijas
import numpy
import math

```

```
def videjais_aritmetiskais(a):
```

```
    garums = len(a)
```

```
    summa = 0
```

```
    for i in range(garums):
```

```
        summa += a[i]
```

```
    return summa / garums
```

```
def videja_kvadratiskā_vertība(a):
```

```
    garums = len(a)
```

```
    summa = 0
```

```
    for i in range(garums):
```

```
        summa += a[i]*a[i]
```

```
    return math.sqrt(summa / garums)
```

```
def videja_harmoniskā_vertība(a):
```

```
    garums = len(a)
```

```
    summa = 0
```

```
    for i in range(garums):
```

```
        if a[i] != 0:
```

```
            summa += 1 / a[i]
```

```
        else:
```

```
            garums -= 1
```

```
    return garums / summa
```

```
def videja_geometriskā_vertība(a):
```

```
    garums = len(a)
```

```
    reizinājums = 1
```

```
    for i in range(garums):
```

```
        reizinājums *= a[i]
```

```
    return reizinājums ** (1/garums)
```

```
def videjas_lineāra_novirze(a):
```

```
    garums = len(a)
```

```
    summa = 0
```

```

x = funkcijas.videjais_aritmetiskais(a)

for i in range(garums):
    summa += abs(a[i] - x)

return summa / garums

def standartnovirze(a):
    garums = len(a)
    summa = 0

    x = funkcijas.videjais_aritmetiskais(a)

    for i in range(garums):
        summa += (a[i] - x) * (a[i] - x)

    return math.sqrt(summa / garums)

n = input('Ievadiet skaitļu skaitu --> ')
n = funkcijas.naturals_skaitlis(n)

saraksts = numpy.arange(n)

for i in range(n):
    skaitlis = input('Ievadiet skaitli --> ')
    skaitlis = funkcijas.naturals_skaitlis(skaitlis)
    saraksts[i] = skaitlis

# print(saraksts)
print(f'Vidējais aritmētiskais: {videjais_aritmetiskais(saraksts)}')
print(f'Vidējais kvadrātiskais: {videja_kvadratiska_vertiba(saraksts)}')
print(f'Vidējais harmoniskais: {videja_harmoniska_vertiba(saraksts)}')
print(f'Vidējais ģeometriskais: {videja_geometriskas_vertiba(saraksts)}')
print(f'Vidējā lineāra novirze: {videjas_lineara_novirze(saraksts)}')
print(f'Standartnovirze: {standartnovirze(saraksts)}')

```

Testa piemērs(1)

```

Ievadiet skaitļu skaitu --> 3
Ievadiet vērtību --> 1
Ievadiet vērtību --> 2
Ievadiet vērtību --> 3
Vidējais aritmētiskais: 2.0
Vidējais kvadrātiskais: 2.160246899469287
Vidējais harmoniskais: 1.6363636363636365
Vidējais ģeometriskais: 1.8171205928321397
Vidējā lineāra novirze: 0.6666666666666666
Standartnovirze: 0.816496580927726

```

### Testa piemērs(2)

```
Ievadiet skaitli --> 34
Ievadiet skaitli --> 5
Ievadiet skaitli --> 12
Ievadiet skaitli --> 87
Ievadiet skaitli --> 3
Ievadiet skaitli --> 5
Vidējais aritmētiskais: 24.333333333333332
Vidējais kvadrātiskais: 38.57460304397182
Vidējais harmoniskais: 6.996491504710845
Vidējais ģeometriskais: 11.772628689396555
Vidējā lineāra novirze: 24.111111111111111
Standartnovirze: 29.931403055802264
```

### Testa piemērs(3)

```
Ievadiet skaitļu skaitu --> 1
Ievadiet skaitli --> 67
Vidējais aritmētiskais: 67.0
Vidējais kvadrātiskais: 67.0
Vidējais harmoniskais: 67.0
Vidējais ģeometriskais: 67.0
Vidējā lineāra novirze: 0.0
Standartnovirze: 0.0
```

## 3.uzdevums

Programma ar funkcijām masīva izveidei, aizpildīšanai ar datiem, izvadei un mazāka un lielāka atrašanai.

### Kods:

Funkcijas no *funkcijas.py*:

```
# Funkcija pārbaudei, lai skaitlis būtu naturāls skaitlis
def naturals_skaitlis(a):
    # meģinājumu skaita skaitītājs (3 meģinājumi ierakstam)
    meģinajumi = 1
    while meģinajumi <= 3:
        try:
            a = int(a)
            if a > 0:
                return int(a)
        except:
            raise Exception
        meģinajumi += 1
    a = input('Ievadiet naturālo skaitli vēlreiz --> ')
    print('Programma beidz darbību!')
    exit()
```

```

# Funkcija pārbaudei, lai skaitlis būtu reāls skaitlis
def reals_skaitlis(a):
    # meģinājumu skaitītājs (3 meģinājumi pareizi ievadīt)
    meģinajumi = 1
    while meģinajumi <= 3:
        try:
            a = float(a)
            return float(a)
        except:
            if meģinajumi < 3:
                meģinajumi += 1
                a = input('Ievadiet reālo skaitli vēlreiz --> ')
            else:
                print('Programma beidz darbību!')
                exit()

```

1MPR04\_3\_Simona\_Blinova.py:

```

import numpy
import funkcijas

```

```

def videja_sverta(x, y):

```

```

    summa1 = 0

```

```

    summa2 = 0

```

```

    for i in range(len(x)):

```

```

        summa1 += x[i]*y[i]

```

```

        summa2 += x[i]

```

```

    return summa1 / summa2

```

```

n = input('Ievadiet masīvu izmēru --> ')

```

```

n = funkcijas.naturals_skaitlis(n)

```

```

vertibas = numpy.arange(n)

```

```

skaits = numpy.arange(n)

```

```

for i in range(n):

```

```

    vertiba = input('Ievadiet vērtību --> ')

```

```

    vertiba = funkcijas.reals_skaitlis(vertiba)

```

```

    vertibas[i] = vertiba

```

```

vertibas_skaits = input('Ievadiet vērtības skaitu --> ')

```

```

vertibas_skaits = funkcijas.naturals_skaitlis(vertibas_skaits)

```

```

skaits[i] = vertibas_skaits

```

```
print(f'Vidēja svērtā vērtība: {videja_sverta(vertibas, skaits)}')
```

Testa piemērs(1)

```
Ievadiet masīvu izmēru --> 4
Ieadiet vērtību --> -3
Ieadiet vērtības skaitu --> 8
Ieadiet vērtību --> 23
Ieadiet vērtības skaitu --> 5
Ieadiet vērtību --> 13
Ieadiet vērtības skaitu --> 1
Ieadiet vērtību --> 4.5
Ieadiet vērtības skaitu --> 5
3.3513513513513513
```

Testa piemērs(2)

```
Ievadiet masīvu izmēru --> 1
Ieadiet vērtību --> 34
Ieadiet vērtības skaitu --> 23
23.0
```

Testa piemērs(3)

```
Ievadiet masīvu izmēru --> 5
Ieadiet vērtību --> -9
Ieadiet vērtības skaitu --> 13
Ieadiet vērtību --> 4
Ieadiet vērtības skaitu --> 1
Ieadiet vērtību --> 12
Ieadiet vērtības skaitu --> 2
Ieadiet vērtību --> -2
Ieadiet vērtības skaitu --> 3
Ieadiet vērtību --> 0
Ieadiet vērtības skaitu --> 5
-19.0
```

#### **4.uzdevums**

Programma ar funkcijām masīva izveidei, aizpildīšanai ar datiem, izvadei un mazāka un lielāka atrašanai.

#### **Kods:**

Funkcijas no *funkcijas.py*:

```
# Funkcija pārbaudei, lai skaitlis būtu naturāls skaitlis
def naturals_skaitlis(a):
    # meģinājumu skaita skaitītājs (3 meģinājumi ierakstam)
    meģinajumi = 1
    while meģinajumi <= 3:
        try:
            a = int(a)
            if a > 0:
```



```

        return int(a)
    else:
        raise Exception
except:
    if meginajumi < 3:
        meginajumi += 1
        a = input('Ievadiet naturālo skaitli vēlreiz --> ')
    else:
        print('Programma beidz darbību!')
        exit()

```

```

# Funkcija pārbaudei, lai skaitlis būtu reāls skaitlis
def reals_skaitlis(a):
    # meģinājumu skaitītājs (3 meģinājumi pareizi ievadīt)
    meginajumi = 1
    while meginajumi <= 3:
        try:
            a = float(a)
            return float(a)
        except:
            if meginajumi < 3:
                meginajumi += 1
                a = input('Ievadiet reālo skaitli vēlreiz --> ')
            else:
                print('Programma beidz darbību!')
                exit()

```

```

# Vidēja aritmētiskā funkcija
def videjais_aritmetiskais(a):
    garums = len(a)
    summa = 0

    for i in range(garums):
        summa += a[i]

    return summa / garums

```

1MPR04\_4\_Simona\_Blinova.py:

```

import funkcijas
import numpy
import math

```

```

def linearas_korelacijas_koeficients(a, b, n):
    x = funkcijas.videjais_aritmetiskais(a)
    y = funkcijas.videjais_aritmetiskais(b)

```

```

summa1 = 0
summa2 = 0
summa3 = 0

for i in range(n):
    summa1 += (a[i] - x) * (b[i] - y)
    summa2 += (a[i] - x) * (a[i] - x)
    summa3 += (b[i] - y) * (b[i] - y)

    if summa2 == 0 or summa3 == 0:
        return 'Kļūda, dalīšana ar nulli'

return summa1 / math.sqrt(summa2 * summa3)

n = input('Ievadiet masīvu izmēru --> ')
n = funkcijas.naturals_skaitlis(n)

noverojumi1 = numpy.arange(n)
noverojumi2 = numpy.arange(n)

for i in range(n):
    skaitlis = input('Ievadiet skaitli 1. masīvā --> ')
    skaitlis = funkcijas.reals_skaitlis(skaitlis)
    noverojumi1[i] = skaitlis

    skaitlis = input('Ievadiet skaitli 2. masīvā --> ')
    skaitlis = funkcijas.reals_skaitlis(skaitlis)
    noverojumi2[i] = skaitlis

print(f'Lineāras korelācijas koeficients: {linearas_korelacijas_koeficients(noverojumi1,
noverojumi2, n)}')

```

Testa piemērs(1)

```

Ievadiet masīvu izmēru --> 5
Ievadiet skaitli 1. masīvā --> 12
Ievadiet skaitli 2. masīvā --> 34
Ievadiet skaitli 1. masīvā --> 12
Ievadiet skaitli 2. masīvā --> -3
Ievadiet skaitli 1. masīvā --> 6
Ievadiet skaitli 2. masīvā --> -4
Ievadiet skaitli 1. masīvā --> 54
Ievadiet skaitli 2. masīvā --> -9
Ievadiet skaitli 1. masīvā --> 7
Ievadiet skaitli 2. masīvā --> 6
-0.38339113124002194

```

### Testa piemērs(2)

```
Ievadiet masīvu izmēru --> 3
Ievadiet skaitli 1. masīvā --> 1
Ievadiet skaitli 2. masīvā --> 4
Ievadiet skaitli 1. masīvā --> 2
Ievadiet skaitli 2. masīvā --> 5
Ievadiet skaitli 1. masīvā --> 3
Ievadiet skaitli 2. masīvā --> 6
1.0
```

### Testa piemērs(3)

```
Ievadiet masīvu izmēru --> 1
Ievadiet skaitli 1. masīvā --> 12
Ievadiet skaitli 2. masīvā --> -4
Kļūda, dalīšana ar nulli
```