

5. praktiskais darbs. 2. semestris

1. uzdevums

Sastādīt programmu, kas pieprasa lietotājam ievadīt N skaitļus un atrod šo skaitļu virknes:

1.1. Vidējā aritmētiskā vērtību, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$x = \frac{\sum_{i=1}^n x_i}{n}$$

1.2. Vidējā kvadrātiskā vērtību, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$x = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}}$$

1.3. Vidējā harmoniskā vērtību, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$x = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

1.4. Vidējā ģeometriskā vērtību, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$x = \sqrt[n]{\prod_{i=1}^n x_i}$$

1.5. Vidējās absolūtās jeb vidējās lineārās novirzes vērtību, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$\alpha = \frac{\sum_{i=1}^n |x_i - x|}{n}, \text{ kur } x = \frac{\sum_{i=1}^n x_i}{n}$$

1.6. Standartnovirzes jeb vidējās kvadrātiskās novirzes vērtību, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - x)^2}{n}}, \text{ kur } x = \frac{\sum_{i=1}^n x_i}{n}$$

Zināšanai, standartnovirzi aprēķina kā kvadrātsakni no dispersijas

Kods:

```
# Programmas nosaukums: Formulas no statistikas
```

```
# 1. uzdevums (1MPR05_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas pieprasa lietotājam ievadīt N skaitļus un atrod šo skaitļu virknes
```

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

```
# Versija 1.0
```

```
import numpy
```

```
import math
```

```
def is_natural(n):
```

```
    # Pārbauda vai simbolu virkne ir naturāls skaitlis vai nav
```

```
    # Ja ir naturāls skaitlis, tad True. Ja nav tad False.
```

```
    # n - simbolu virkne, kuru pārbauda.
```

```
    if str(n).isdigit() and float(n) == int(n) and int(n) > 0:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def izveidot_masivu_ar_garumu(n):
```

```
    # Izveido masīvu ar norādīto garumu n
```

```
    # n - naturāls skaitlis
```

```
    a = numpy.arange(n)
```

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

```
        b = input("Ievadiet " + str(i) + ".elementu ==> ")
```

```
        b = is_whole(b, i)
```

```
        a[i] = b
```

```
    return a
```

```
def is_whole(x, i): # Bezgalīgi daudz reizes ievāda
```

```
    while True:
```

```
try:
    x = int(x)
except:
    x = input("Kļūda! Ievadiet " + str(i) + ".elementu ==> ")
else:
    return int(x)
```

```
def izvade(x):
    # Izvada masīva elementus pēc kārtas līdz pedējam
    # x - masīvs
    n = len(x)
    s = str(x[0])
    for i in range(1, n):
        s = s + ", " + str(x[i])
    return s
```

```
def videjais_aritmetiskais(x):
    # Aprēķina masīva vidējo aritmētisko
    # x - masīvs
    n = len(x)
    t = 0
    for i in range(0, n):
        t = t + x[i]

    return t / n
```

```
def videjais_kvadratiskais(x):
    # Aprēķina masīva videjo kvadrātisko
```

```
# x - masīvs
n = len(x)
t = 0
for i in range(0, n):
    t = t + x[i] * x[i]

return math.sqrt(t / n)
```

```
def videjais_harmoniskais(x):
    # Aprēķina masīva videjo harmonisko
    # x - masīvs
    n = len(x)
    t = 0

    for i in range(0, n):

        if x[i] == 0:
            return "Kļūda! Dalīšana ar 0"

        elif is_natural(x[i]):
            t = t + 1 / x[i]

        else:
            return "Kļūda! Visiem skaitļiem jābut pozitīviem!"

    return n / t
```

```
def videjais_geometriskais(x):
    # Aprēķina masīva videjo ģeometrisku
    # x - masīvs
    n = len(x)
```

```

s = 1

for i in x:
    s = s * i

if n % 2 == 0: # Pārbauda vai n-sakne ir pāra skaitlis
    if s >= 0: # Ja n-sakne pāra skaitlis, tad pārbaudam vai nav negatīvs, ja ir tad nevaram
    aprēķināt
        k = math.pow(s, (1 / n)) # ja ir pozitīvs vai 0, tad viss ir labi, varam aprēķināt pāra-sakni
        no pozitīvas vērtības
    else:
        k = "Nevar aprēķināt reālos skaitļos" # ja ir negatīvs un n-sakne ir pāra skaitlis, tad
        nevaram to aprēķināt reālos skaitļos
    else:
        k = numpy.sign(s) * (numpy.abs(s)) ** (1 / n) # ja n-sakne ir nepāra skaitlis, tad aprēķināt
        to šādi (parasta pow(a,b) funkcija nedarbojas ar tādiem skaitļiem)

return k

```

```

def videjas_linearas_novirzes_vertiba(x):
    # Aprēķina masīva vidējās absolūtās jeb vidējās lineārās novirzes vērtību
    # x - masīvs
    n = len(x)
    t = 0
    k = 0
    for i in range(0, n):
        t = t + x[i]

    c = t / n

    for i in range(0, n):

```

```
k = k + abs(x[i] - c)
```

```
return k / n
```

```
def standartnovirze(x):
```

```
    # Aprēķina masīva standartnovirzi
```

```
    # x - masīvs
```

```
    n = len(x)
```

```
    t = 0
```

```
    k = 0
```

```
    for i in range(0, n):
```

```
        t = t + x[i]
```

```
    c = t / n
```

```
    for i in range(0, n):
```

```
        k = k + (x[i] - c) * (x[i] - c)
```

```
    return math.sqrt(k / n)
```

```
# -----
```

```
# Galvenā programmas daļa
```

```
# -----
```

```
n = input("Ievadiet masīva izmēru N ==> ")
```

```
while is_natural(n) == False:
```

```
    n = input("Masīva izmērs ir naturāls skaitlis!\nIevadiet masīva izmēru N ==> ")
```

```

n = izveidot_masivu_ar_garumu(int(n))

print("\nSkaitļu virknes " + izvade(n))

print("Vidējais aritmētiskais: " + str(videjais_aritmetiskais(n)))

print("Vidējais kvadratiskais: " + str(videjais_kvadratiskais(n)))

print("Vidējais harmoniskais: " + str(videjais_harmoniskais(n)))

print("Vidējais ģeometriskais: " + str(videjais_geometriskais(n)))

print("Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: " +
str(videjas_linearas_novirzes_vertiba(n)))

print("Standartnovirze: " + str(standartnovirze(n)))

```

Testa piemēri:

1)

```

Ievadiet masīva izmēru N ==> 3
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3

Skaitļu virknes 1, 2, 3
Vidējais aritmētiskais: 2.0
Vidējais kvadratiskais: 2.160246899469287
Vidējais harmoniskais: 1.6363636363636365
Vidējais ģeometriskais: 1.8171205928321397
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 0.6666666666666666
Standartnovirze: 0.816496580927726

```

2)

```

Ievadiet masīva izmēru N ==> 3
Ievadiet 0.elementu ==> -3
Ievadiet 1.elementu ==> -1
Ievadiet 2.elementu ==> 3

Skaitļu virknes -3, -1, 3
Vidējais aritmētiskais: -0.3333333333333333
Vidējais kvadratiskais: 2.516611478423583
Vidējais harmoniskais: Kļūda! Visiem skaitļiem jābut pozitīviem!
Vidējais ģeometriskais: 2.080083823051904
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 2.2222222222222222
Standartnovirze: 2.494438257849294

```

3)

```
Ievadiet masīva izmēru N ==> 2
Ievadiet 0.elementu ==> 2
Ievadiet 1.elementu ==> 2

Skaitļu virknes 2, 2
Vidējais aritmētiskais: 2.0
Vidējais kvadrātiskais: 2.0
Vidējais harmoniskais: 2.0
Vidējais ģeomētriskais: 2.0
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 0.0
Standartnovirze: 0.0
```

4)

```
Ievadiet masīva izmēru N ==> 2
Ievadiet 0.elementu ==> -2
Ievadiet 1.elementu ==> 1

Skaitļu virknes -2, 1
Vidējais aritmētiskais: -0.5
Vidējais kvadrātiskais: 1.5811388300841898
Vidējais harmoniskais: Kļūda! Visiem skaitļiem jābut pozitīviem!
Vidējais ģeomētriskais: Nevar aprēķināt reālos skaitļos
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 1.5
Standartnovirze: 1.5
```

5)

```
Ievadiet masīva izmēru N ==> 6
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 5
Ievadiet 5.elementu ==> 6

Skaitļu virknes 1, 2, 3, 4, 5, 6
Vidējais aritmētiskais: 3.5
Vidējais kvadrātiskais: 3.8944404818493075
Vidējais harmoniskais: 2.4489795918367347
Vidējais ģeomētriskais: 2.993795165523909
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 1.5
Standartnovirze: 1.707825127659933
```


6)

```
Ievadiet masīva izmēru N ==> 6
Ievadiet 0.elementu ==> -1
Ievadiet 1.elementu ==> -2
Ievadiet 2.elementu ==> -3
Ievadiet 3.elementu ==> -4
Ievadiet 4.elementu ==> -5
Ievadiet 5.elementu ==> -6

Skaitļu virknes -1, -2, -3, -4, -5, -6
Vidējais aritmētiskais: -3.5
Vidējais kvadrātiskais: 3.8944404818493075
Vidējais harmoniskais: Kļūda! Visiem skaitļiem jābut pozitīviem!
Vidējais ģeomētriskais: 2.993795165523909
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 1.5
Standartnovirze: 1.707825127659933
```

7)

```
Ievadiet masīva izmēru N ==> 2
Ievadiet 0.elementu ==> 0
Ievadiet 1.elementu ==> 0

Skaitļu virknes 0, 0
Vidējais aritmētiskais: 0.0
Vidējais kvadrātiskais: 0.0
Vidējais harmoniskais: Kļūda! Dalīšana ar 0
Vidējais ģeomētriskais: 0.0
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 0.0
Standartnovirze: 0.0
```

8)

```
Ievadiet masīva izmēru N ==> 6
Ievadiet 0.elementu ==> -2
Ievadiet 1.elementu ==> 100
Ievadiet 2.elementu ==> 4
Ievadiet 3.elementu ==> 3
Ievadiet 4.elementu ==> -1
Ievadiet 5.elementu ==> 6

Skaitļu virknes -2, 100, 4, 3, -1, 6
Vidējais aritmētiskais: 18.333333333333332
Vidējais kvadrātiskais: 40.959329421594134
Vidējais harmoniskais: Kļūda! Visiem skaitļiem jābut pozitīviem!
Vidējais ģeomētriskais: 4.93242414866094
Vidējās absolūtās jeb vidējās lineārās novirzes vērtība: 27.222222222222225
Standartnovirze: 36.62725154247252
```

2. uzdevums

Sastādīt programmu, kas pieprasa lietotājam ievadīt divus vienāda izmēra datu masīvus ar vērtībām un to skaitu un aprēķina vidējā svēta vērtību, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$x = \frac{\sum_{i=1}^n x_i \cdot y_i}{\sum_{i=1}^n y_i}$$

Kods:

```
# Programmas nosaukums: Vidēja svēta vērtība
```

```
# 2. uzdevums (1MPR05_Vladislavs_Babaņins)
```

Uzdevuma formulējums: Sastādīt programmu, kas pieprasa lietotājam ievadīt divus vienāda izmēra datu masīvus ar novērojumiem un aprēķina lineārās korelācijas koeficienta vērtību starp šo abu datu masīvu atbilstošiem elementiem, ja zināms, ka to var aprēķināt pēc šādas formulas:

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

```
# Versija 1.0
```

```
import numpy
```

```
import math
```

```
def is_natural(n):
```

```
    # Pārbauda vai simbolu virkne ir naturāls skaitlis vai nav
```

```
    # Ja ir naturāls skaitlis, tad True. Ja nav tad False.
```

```
    # n - simbolu virkne, kuru pārbauda.
```

```
    if str(n).isdigit() and float(n) == int(n) and int(n) > 0:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def izveidot_masivu_ar_garumu(n):
```

```
# Izveido masīvu ar norādīto garumu n

# n - naturāls skaitlis

a = numpy.arange(n)

for i in range(n):

    b = input("Ievadiet " + str(i) + ".elementu ==> ")

    b = is_whole(b, i)

    a[i] = b

return a
```

```
def is_whole(x, i): # Bezgalīgi daudz reizes ievāda

    while True:

        try:

            x = int(x)

        except:

            x = input("Kļūda! Ievadiet " + str(i) + ".elementu ==> ")

        else:

            return int(x)
```

```
def izvade(x):

    # Izvada masīva elementus pēc kārtas līdz pedējam

    # x - masīvs

    n = len(x)

    s = str(x[0])

    for i in range(1, n):

        s = s + ", " + str(x[i])

    print(s)
```

```
def videjais_aritmetiskais(x):
```

```
# Aprēķina masīva vidējo aritmētisko
```

```
# x - masīvs
```

```
n = len(x)
```

```
t = 0
```

```
for i in range(0, n):
```

```
    t = t + x[i]
```

```
return t / n
```

```
def videja_sversta_vertiba(x, y):
```

```
    # Aprēķina masīvu vidējo svērsto vērtību
```

```
    # x - pirmais masīvs
```

```
    # y - otrais masīvs
```

```
n = len(x)
```

```
t = 0
```

```
z = 0
```

```
for i in range(0, n):
```

```
    t = t + x[i] * y[i]
```

```
for i in range(0, n):
```

```
    z = z + y[i]
```

```
return t / z
```

```
# -----
```

```
# Galvenā programmas daļa
```

```
# -----
```

```

m = input("Ievadiet masīva izmēru N ==> ")

while is_natural(m) == False:
    m = input("Masīva izmērs ir naturāls skaitlis!\nIevadiet masīva izmēru N ==> ")

m = int(m)
t = izveidot_masivu_ar_garumu(m)
print("Ievadiet elementu svarus!")
c = izveidot_masivu_ar_garumu(m)

print("\nPirmas virknes elementi:")
izvade(t)
print("Otrās virknes elementi (svars):")
izvade(c)
print("Vidēja svērtā vērtība ir:")
print(videja_sversta_vertiba(t, c))

```

Testa piemēri:

1)

```

Ievadiet masīva izmēru N ==> 3
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet elementu svarus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3

Pirmas virknes elementi:
1, 2, 3
Otrās virknes elementi (svars):
1, 2, 3
Vidēja svērtā vērtība ir:
2.3333333333333335

```

2)

```
Ievadiet masīva izmēru N ==> 0
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> -7
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> pieci
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 1
Ievadiet 0.elementu ==> 5
Ievadiet elementu svarus!
Ievadiet 0.elementu ==> 5

Pirmas virknes elementi:
5
Otrās virknes elementi (svars):
5
Vidēja svērtā vērtība ir:
5.0
```

3)

```
Ievadiet masīva izmēru N ==> 3
Ievadiet 0.elementu ==> Labi!
Kļūda! Ievadiet 0.elementu ==> 5
Ievadiet 1.elementu ==> 5
Ievadiet 2.elementu ==> 5
Ievadiet elementu svarus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 1
Ievadiet 2.elementu ==> 1

Pirmas virknes elementi:
5, 5, 5
Otrās virknes elementi (svars):
1, 1, 1
Vidēja svērtā vērtība ir:
5.0
```

4)

```
Ievadiet masīva izmēru N ==> 3
Ievadiet 0.elementu ==> 9
Ievadiet 1.elementu ==> 8
Ievadiet 2.elementu ==> 10
Ievadiet elementu svarus!
Ievadiet 0.elementu ==> 8
Ievadiet 1.elementu ==> 4
Ievadiet 2.elementu ==> 2
```

Pirmas virknes elementi:

9, 8, 10

Otrās virknes elementi (svars):

8, 4, 2

Vidēja svērtā vērtība ir:

8.857142857142858

5)

```
Ievadiet masīva izmēru N ==> 6
Ievadiet 0.elementu ==> 9
Ievadiet 1.elementu ==> 7
Ievadiet 2.elementu ==> 10
Ievadiet 3.elementu ==> 10
Ievadiet 4.elementu ==> 4
Ievadiet 5.elementu ==> 6
Ievadiet elementu svarus!
Ievadiet 0.elementu ==> 6
Ievadiet 1.elementu ==> 4
Ievadiet 2.elementu ==> 2
Ievadiet 3.elementu ==> 2
Ievadiet 4.elementu ==> 12
Ievadiet 5.elementu ==> 2
```

Pirmas virknes elementi:

9, 7, 10, 10, 4, 6

Otrās virknes elementi (svars):

6, 4, 2, 2, 12, 2

Vidēja svērtā vērtība ir:

6.5

6)

```
Ievadiet masīva izmēru N ==> -7
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 7
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 5
Ievadiet 5.elementu ==> 6
Ievadiet 6.elementu ==> 7
Ievadiet elementu svarus!
Ievadiet 0.elementu ==> 10000
Ievadiet 1.elementu ==> 1
Ievadiet 2.elementu ==> 1
Ievadiet 3.elementu ==> 1
Ievadiet 4.elementu ==> 2
Ievadiet 5.elementu ==> 3
Ievadiet 6.elementu ==> 5

Pirmas virknes elementi:
1, 2, 3, 4, 5, 6, 7
Otrās virknes elementi (svars):
10000, 1, 1, 1, 2, 3, 5
Vidēja svērtā vērtība ir:
1.0058923399580546
```

7)

```
Ievadiet masīva izmēru N ==> 0
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 1
Ievadiet 0.elementu ==> 5
Ievadiet elementu svarus!
Ievadiet 0.elementu ==> 88888

Pirmas virknes elementi:
5
Otrās virknes elementi (svars):
88888
Vidēja svērtā vērtība ir:
5.0
```


3. uzdevums

Sastādīt programmu, kas pieprasa lietotājam ievadīt divus vienāda izmēra datu masīvus ar novērojumiem un aprēķina lineārās korelācijas koeficienta vērtību starp šo abu datu masīvi atbilstošiem elementiem, ja zināms, ka to var aprēķināt pēc šādas formulas:

$$k = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2}}, \text{ kur } \bar{x} = \frac{\sum_{i=1}^n x_i}{n}, \bar{y} = \frac{\sum_{i=1}^n y_i}{n},$$

Kods:

```
# Programmas nosaukums: Lineārās korelācijas koeficienta vērtība
```

```
# 3. uzdevums (1MPR05_Vladislavs_Babaņins)
```

Uzdevuma formulējums: Sastādīt programmu, kas pieprasa lietotājam ievadīt divus vienāda izmēra datu masīvus ar novērojumiem un aprēķina lineārās korelācijas koeficienta vērtību starp šo abu datu masīvi atbilstošiem elementiem, ja zināms, ka to var aprēķināt pēc šādas formulas:

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

```
# Versija 1.0
```

```
import numpy
```

```
import math
```

```
def is_natural(n):
```

```
    # Pārbauda vai simbolu virkne ir naturāls skaitlis vai nav
```

```
    # Ja ir naturāls skaitlis, tad True. Ja nav tad False.
```

```
    # n - simbolu virkne, kuru pārbauda.
```

```
    if str(n).isdigit() and float(n) == int(n) and int(n) > 0:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def izveidot_masivu_ar_garumu(n):
```

```
# Izveido masīvu ar norādīto garumu n

# n - naturāls skaitlis

a = numpy.arange(n)

for i in range(n):

    b = input("Ievadiet " + str(i) + ".elementu ==> ")

    b = is_whole(b, i)

    a[i] = b

return a
```

```
def is_whole(x, i): # Bezgalīgi daudz reizes ievāda

    while True:

        try:

            x = int(x)

        except:

            x = input("Kļūda! Ievadiet " + str(i) + ".elementu ==> ")

        else:

            return int(x)
```

```
def izvade(x):

    # Izvada masīva elementus pēc kārtas līdz pedējam

    # x - masīvs

    n = len(x)

    s = str(x[0])

    for i in range(1, n):

        s = s + ", " + str(x[i])

    print(s)
```

```
def videjais_aritmetiskais(x):
```

```
# Aprēķina masīva vidējo aritmētisko
```

```
# x - masīvs
```

```
n = len(x)
```

```
t = 0
```

```
for i in range(0, n):
```

```
    t = t + x[i]
```

```
return t / n
```

```
def linearas_korelacijas_koeficients(x, y):
```

```
    # Aprēķina masīvu linearas korelacijas koeficientu
```

```
    # x - pirmais masīvs
```

```
    # y - otrais masīvs
```

```
    n = len(x)
```

```
    vidx = videjais_aritmetiskais(x)
```

```
    vidy = videjais_aritmetiskais(y)
```

```
    s = 0
```

```
    tx = 0
```

```
    ty = 0
```

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

```
        s = s + (x[i] - vidx) * (y[i] - vidy)
```

```
        tx = tx + (x[i] - vidx) * (x[i] - vidx)
```

```
        ty = ty + (y[i] - vidy) * (y[i] - vidy)
```

```
    p = math.sqrt(tx * ty)
```

```
    if p != 0:
```

```
        return s / p
```

else:

return "Kļūda! Dalīšana ar 0"

Galvenā programmas daļa

m = input("Ievadiet masīva izmēru N ==> ")

while is_natural(m) == False:

m = input("Masīva izmērs ir naturāls skaitlis!\nIevadiet masīva izmēru N ==> ")

m = int(m)

print("Ievadiet pirmā masīva skaitļus!")

t = izveidot_masivu_ar_garumu(m)

print("Ievadiet otrā masīva skaitļus!")

c = izveidot_masivu_ar_garumu(m)

print("\nPirmā skaitļu virkne:")

izvade(t)

print("Otrā skaitļu virkne:")

izvade(c)

print("Lineāras korelācijas koeficients:")

print(linearas_korelacijas_koeficients(t, c))

Testa piemēri:

1)

```
Ievadiet masīva izmēru N ==> 3
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3

Pirmā skaitļu virkne:
1, 2, 3
Otrā skaitļu virkne:
1, 2, 3
Lineāras korelācijas koeficients:
1.0
```

2)

```
Ievadiet masīva izmēru N ==> pieci
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 0
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> -2
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 3
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> -1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 5
Ievadiet 2.elementu ==> 8

Pirmā skaitļu virkne:
-1, 2, 3
Otrā skaitļu virkne:
1, 5, 8
Lineāras korelācijas koeficients:
0.9803156180632394
```

3)

```
Ievadiet masīva izmēru N ==> 3
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 66
Ievadiet 1.elementu ==> 512
Ievadiet 2.elementu ==> 23

Pirmā skaitļu virkne:
1, 2, 3
Otrā skaitļu virkne:
66, 512, 23
Lineāras korelācijas koeficients:
-0.07940429881402113
```

4)

```
Ievadiet masīva izmēru N ==> 5
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 10
Ievadiet 1.elementu ==> 9
Ievadiet 2.elementu ==> 8
Ievadiet 3.elementu ==> 7
Ievadiet 4.elementu ==> 6
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 5

Pirmā skaitļu virkne:
10, 9, 8, 7, 6
Otrā skaitļu virkne:
1, 2, 3, 4, 5
Lineāras korelācijas koeficients:
-1.0
```

5)

```
Ievadiet masīva izmēru N ==> 10
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 5
Ievadiet 5.elementu ==> 6
Ievadiet 6.elementu ==> 7
Ievadiet 7.elementu ==> 8
Ievadiet 8.elementu ==> 9
Ievadiet 9.elementu ==> 10
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 0
Ievadiet 1.elementu ==> 0
Ievadiet 2.elementu ==> 0
Ievadiet 3.elementu ==> 0
Ievadiet 4.elementu ==> 0
Ievadiet 5.elementu ==> 0
Ievadiet 6.elementu ==> 0
Ievadiet 7.elementu ==> 0
Ievadiet 8.elementu ==> 0
Ievadiet 9.elementu ==> 0
```

Pirmā skaitļu virkne:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Otrā skaitļu virkne:

0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Lineāras korelācijas koeficients:

Kļūda! Dalīšana ar 0

6)

```
Ievadiet masīva izmēru N ==> 3
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 0
Ievadiet 1.elementu ==> 0
Ievadiet 2.elementu ==> 0
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 0
Ievadiet 1.elementu ==> 0
Ievadiet 2.elementu ==> 0
```

Pirmā skaitļu virkne:

0, 0, 0

Otrā skaitļu virkne:

0, 0, 0

Lineāras korelācijas koeficients:

Kļūda! Dalīšana ar 0

7)

```
Ievadiet masīva izmēru N ==> 3
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 10
Ievadiet 1.elementu ==> 9
Ievadiet 2.elementu ==> 8
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 7
Ievadiet 1.elementu ==> 6
Ievadiet 2.elementu ==> 5
```

Pirmā skaitļu virkne:

10, 9, 8

Otrā skaitļu virkne:

7, 6, 5

Lineāras korelācijas koeficients:

1.0

8)

```
Ievadiet masīva izmēru N ==> 3
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 1000
Ievadiet 1.elementu ==> 10
Ievadiet 2.elementu ==> 99
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 5
Ievadiet 1.elementu ==> 6
Ievadiet 2.elementu ==> 777

Pirmā skaitļu virkne:
1000, 10, 99
Otrā skaitļu virkne:
5, 6, 777
Lineāras korelācijas koeficients:
-0.4289969886021392
```

9)

```
Ievadiet masīva izmēru N ==> 5
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 7
Ievadiet 1.elementu ==> 7
Ievadiet 2.elementu ==> 7
Ievadiet 3.elementu ==> 1
Ievadiet 4.elementu ==> 1
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 1
Ievadiet 2.elementu ==> 8
Ievadiet 3.elementu ==> 9
Ievadiet 4.elementu ==> 1

Pirmā skaitļu virkne:
7, 7, 7, 1, 1
Otrā skaitļu virkne:
1, 1, 8, 9, 1
Lineāras korelācijas koeficients:
-0.22140372138502384
```

10)

```
Ievadiet masīva izmēru N ==> 1
Ievadiet pirmā masīva skaitļus!
Ievadiet 0.elementu ==> 1
Ievadiet otrā masīva skaitļus!
Ievadiet 0.elementu ==> 1

Pirmā skaitļu virkne:
1
Otrā skaitļu virkne:
1
Lineāras korelācijas koeficients:
Kļūda! Dalīšana ar 0
```

4. uzdevums

Sastādīt programmu, kas pieprasa lietotājam ievadīt N skaitļus un sakārto tos dilstošā (neaugošā) secībā, izmantojot **naivās** kārtošanas metodi un uz ekrāna izvada sakārtoto skaitļu masīvu un veikto salīdzināšanas skaitu.

NB! Jāraksta programma, kas veic sakārtošanu tieši dilstošā (neaugošā) secībā! Nedrīkst vispirms sakārtot augošā (nedilstošā) secībā un tad nodrukāt masīvu pretējā secībā.

Kods:

```
# Programmas nosaukums: Naivās kārtošanas metode
```

```
# 4. uzdevums (1MPR05_Vladislavs_Babaņins)
```

Uzdevuma formulējums: Sastādīt programmu, kas pieprasa lietotājam ievadīt N skaitļus un sakārto tos dilstošā (neaugošā) secībā, izmantojot naivās kārtošanas metodi un uz ekrāna izvada sakārtoto skaitļu masīvu un veikto salīdzināšanas skaitu.

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

```
# Versija 1.0
```

```
import numpy
```

```
def is_natural(n):
```

```
# Pārbauda vai simbolu virkne ir naturāls skaitlis vai nav
# Ja ir naturāls skaitlis, tad True. Ja nav tad False.
# n - simbolu virkne, kuru pārbauda.
if str(n).isdigit() and float(n) == int(n) and int(n) > 0:
    return True
else:
    return False
```

```
def izvade(x):
    # Izvada masīva elementus pēc kārtas līdz pedējam
    # x - masīvs
    n = len(x)
    s = str(x[0])
    for i in range(1, n):
        s = s + ", " + str(x[i])
    print(s)
```

```
def izveidot_masivu_ar_garumu(n):
    # Izveido masīvu ar norādīto garumu n
    # n - naturāls skaitlis
    a = numpy.arange(n)
    for i in range(n):
        b = input("Ievadiet " + str(i) + ".elementu ==> ")
        b = is_whole(b, i)
        a[i] = b
    return a
```

```
def is_whole(x, i): # Bezgalīgi daudz reizes ievāda
```

```

# Pārbauda vai x ir vesels skaitlis

# x - pārbaudama simbolu virkne

# i - elements pēc kārtas

while True:

    try:

        x = int(x)

    except:

        x = input("Kļūda! Ievadiet " + str(i) + ".elementu ==> ")

    else:

        return int(x)

```

```

def naivais_dilstosa(a):

    # Naivā kārtošanas metode

    # Sakarto masīva elementus dilstoša (neaugoša) secība un

    # izvada paveikto salīdzināšanas skaits, lai sakārtot masīvu. (izmanto naivo kārtošanas
metode)

    # a - masīvs

    skaititajs = 0

    n = len(a)

    for j in range(n - 1):

        min1 = a[j]

        imin = j

        for i in range(j + 1, n):

            if min1 < a[i]:

                skaititajs = skaititajs + 1 # skaititājs

                min1 = a[i]

                imin = i

        a[imin] = a[j]

        a[j] = min1

```

```

print(skaititajs) # Izvada paveikto salīdzināšanas skaits, lai sakārtot masīvu

# -----
# Galvenā programmas daļa
# -----

m = input("Ievadiet masīva izmēru N ==> ")

while is_natural(m) == False:
    m = input("Masīva izmērs ir naturāls skaitlis!\nIevadiet masīva izmēru N ==> ")

m = int(m)
b = izveidot_masivu_ar_garumu(m)

print("Paveikts salīdzināšanas skaits, lai sakārtot masīvu:")
naivais_dilstosa(b) # Salīdzināšanas skaits
print("Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:")
izvade(b)

```

Testa piemēri:

1)

```

Ievadi masīva izmēru ==> 3
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
2
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
3, 2, 1

```

2)

```
Ievadi masīva izmēru ==> 12
Ievadiet 0.elementu ==> 65
Ievadiet 1.elementu ==> 4121
Ievadiet 2.elementu ==> 61
Ievadiet 3.elementu ==> 6
Ievadiet 4.elementu ==> 891
Ievadiet 5.elementu ==> 32
Ievadiet 6.elementu ==> 1879
Ievadiet 7.elementu ==> 1
Ievadiet 8.elementu ==> 20
Ievadiet 9.elementu ==> 0
Ievadiet 10.elementu ==> 289
Ievadiet 11.elementu ==> 5
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
13
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
4121, 1879, 891, 289, 65, 61, 32, 20, 6, 5, 1, 0
```

3)

```
Ievadi masīva izmēru ==> 1
Ievadiet 0.elementu ==> 1
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
0
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
1
```

4)

```
Ievadi masīva izmēru ==> 5
Ievadiet 0.elementu ==> 5
Ievadiet 1.elementu ==> 4
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 2
Ievadiet 4.elementu ==> 1
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
0
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
5, 4, 3, 2, 1
```

5)

```
Ievadi masīva izmēru ==> 25
Ievadiet 0.elementu ==> 9
Ievadiet 1.elementu ==> 4
Ievadiet 2.elementu ==> 61
Ievadiet 3.elementu ==> 53
Ievadiet 4.elementu ==> 4698
Ievadiet 5.elementu ==> 47
Ievadiet 6.elementu ==> 15
Ievadiet 7.elementu ==> 3
Ievadiet 8.elementu ==> 21
Ievadiet 9.elementu ==> 35
Ievadiet 10.elementu ==> 4
Ievadiet 11.elementu ==> 9
Ievadiet 12.elementu ==> 47
Ievadiet 13.elementu ==> 23
Ievadiet 14.elementu ==> 13
Ievadiet 15.elementu ==> 8
Ievadiet 16.elementu ==> 49
Ievadiet 17.elementu ==> 1
Ievadiet 18.elementu ==> 3
Ievadiet 19.elementu ==> 516
Ievadiet 20.elementu ==> 49
Ievadiet 21.elementu ==>
Kļūda! Ievadiet 21.elementu ==> 2
Ievadiet 22.elementu ==> 4
Ievadiet 23.elementu ==> 94
Ievadiet 24.elementu ==> 98
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
42
Sakārtots skaitļu masīvs dilstoša (neaugoša) secībā:
4698, 516, 98, 94, 61, 53, 49, 49, 47, 47, 35, 23, 21, 15, 13, 9, 9, 8, 4, 4, 4, 3, 3, 2, 1
```

6)

```
Ievadi masīva izmēru ==> 25
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 5
Ievadiet 5.elementu ==> 6
Ievadiet 6.elementu ==> 7
Ievadiet 7.elementu ==> 8
Ievadiet 8.elementu ==> 9
Ievadiet 9.elementu ==> 10
Ievadiet 10.elementu ==> 11
Ievadiet 11.elementu ==> 12
Ievadiet 12.elementu ==> 13
Ievadiet 13.elementu ==> 14
Ievadiet 14.elementu ==> 15
Ievadiet 15.elementu ==> 16
Ievadiet 16.elementu ==> 17
Ievadiet 17.elementu ==> 18
Ievadiet 18.elementu ==> 19
Ievadiet 19.elementu ==> 20
Ievadiet 20.elementu ==> 21
Ievadiet 21.elementu ==> 22
Ievadiet 22.elementu ==> 23
Ievadiet 23.elementu ==> 24
Ievadiet 24.elementu ==> 25
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
156
Sakārtots skaitļu masīvs dilstoša (neaugoša) secībā:
25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

7)

```
Ievadi masīva izmēru ==> 25
Ievadiet 0.elementu ==> 25
Ievadiet 1.elementu ==> 24
Ievadiet 2.elementu ==> 23
Ievadiet 3.elementu ==> 22
Ievadiet 4.elementu ==> 21
Ievadiet 5.elementu ==> 20
Ievadiet 6.elementu ==> 19
Ievadiet 7.elementu ==> 18
Ievadiet 8.elementu ==> 17
Ievadiet 9.elementu ==> 16
Ievadiet 10.elementu ==> 15
Ievadiet 11.elementu ==> 14
Ievadiet 12.elementu ==> 13
Ievadiet 13.elementu ==> 12
Ievadiet 14.elementu ==> 11
Ievadiet 15.elementu ==> 10
Ievadiet 16.elementu ==> 9
Ievadiet 17.elementu ==> 8
Ievadiet 18.elementu ==> 7
Ievadiet 19.elementu ==> 6
Ievadiet 20.elementu ==> 5
Ievadiet 21.elementu ==> 4
Ievadiet 22.elementu ==> 3
Ievadiet 23.elementu ==> 2
Ievadiet 24.elementu ==> 1
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
0
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

8)

```
Ievadi masīva izmēru ==> 5
Ievadiet 0.elementu ==> -2
Ievadiet 1.elementu ==> 5
Ievadiet 2.elementu ==> 5
Ievadiet 3.elementu ==> 15.3
Kļūda! Ievadiet 3.elementu ==> 12.3
Kļūda! Ievadiet 3.elementu ==> pieci
Kļūda! Ievadiet 3.elementu ==> 5
Ievadiet 4.elementu ==> 6
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
2
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
6, 5, 5, 5, -2
```


9)

```
Ievadi masīva izmēru ==> 10
Ievadiet 0.elementu ==> 2
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 2
Ievadiet 3.elementu ==> 2
Ievadiet 4.elementu ==> 2
Ievadiet 5.elementu ==> 2
Ievadiet 6.elementu ==> 2
Ievadiet 7.elementu ==> 2
Ievadiet 8.elementu ==> 2
Ievadiet 9.elementu ==> 2
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
0
Sakārtots skaitļu masīvs dilstošā (neaugoša) secībā:
2, 2, 2, 2, 2, 2, 2, 2, 2, 2
```

10)

```
Ievadi masīva izmēru ==> 10
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 5
Ievadiet 5.elementu ==> 6
Ievadiet 6.elementu ==> 7
Ievadiet 7.elementu ==> 8
Ievadiet 8.elementu ==> 9
Ievadiet 9.elementu ==> 10
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
25
Sakārtots skaitļu masīvs dilstošā (neaugoša) secībā:
10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

11)

```
Ievadi masīva izmēru ==> 10
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 6
Ievadiet 2.elementu ==> 5
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 2
Ievadiet 5.elementu ==> 3
Ievadiet 6.elementu ==> 8
Ievadiet 7.elementu ==> 9
Ievadiet 8.elementu ==> 10
Ievadiet 9.elementu ==> 1
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
13
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
10, 9, 8, 6, 5, 4, 3, 2, 1, 1
```

12)

```
Ievadi masīva izmēru ==> 10
Ievadiet 0.elementu ==> 10
Ievadiet 1.elementu ==> 9
Ievadiet 2.elementu ==> 8
Ievadiet 3.elementu ==> 7
Ievadiet 4.elementu ==> 6
Ievadiet 5.elementu ==> 5
Ievadiet 6.elementu ==> 4
Ievadiet 7.elementu ==> 3
Ievadiet 8.elementu ==> 2
Ievadiet 9.elementu ==> 1
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
0
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

13)

```
Ievadiet masīva izmēru N ==> -3
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> pieci
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 3
Ievadiet 0.elementu ==> pieci
Kļūda! Ievadiet 0.elementu ==> pieci
Kļūda! Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
2
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
3, 2, 1
```

14)

```
Ievadiet masīva izmēru N ==> 7
Ievadiet 0.elementu ==> -12.2
Kļūda! Ievadiet 0.elementu ==> 12
Ievadiet 1.elementu ==> p
Kļūda! Ievadiet 1.elementu ==> 12
Ievadiet 2.elementu ==> 34
Ievadiet 3.elementu ==> 12
Ievadiet 4.elementu ==> -3
Ievadiet 5.elementu ==> 444
Ievadiet 6.elementu ==> 121
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
6
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
444, 121, 34, 12, 12, 12, -3
```

5. uzdevums

Sastādīt programmu, kas pieprasa lietotājam ievadīt N skaitļus un sakārto tos dilstošā (neaugošā) secībā, izmantojot **burbuļa** kārtošanas metodi un uz ekrāna izvada sakārtoto skaitļu masīvu un veikto salīdzināšanas skaitu.

NB! Jāraksta programma, kas veic sakārtošanu tieši dilstošā (neaugošā) secībā! Nedrīkst vispirms sakārtot augošā (nedilstošā) secībā un tad nodrukāt masīvu pretējā secībā.

Kods:

```
# Programmas nosaukums: Burbuļa kārtošanas metode
```

```
# 5. uzdevums (1MPR05_Vladislavs_Babaņins)
```

Uzdevuma formulējums: Sastādīt programmu, kas pieprasa lietotājam ievadīt N skaitļus un sakārto tos dilstošā (neaugošā) secībā, izmantojot burbuļa kārtošanas metodi un uz ekrāna izvada sakārtoto skaitļu masīvu un veikto salīdzināšanas skaitu.

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

```
# Versija 1.0
```

```
import numpy
```

```
def is_natural(n):
```

```
    # Pārbauda vai simbolu virkne ir naturāls skaitlis vai nav
```

```
    # Ja ir naturāls skaitlis, tad True. Ja nav tad False.
```

```
    # n - simbolu virkne, kuru pārbauda.
```

```
    if str(n).isdigit() and float(n) == int(n) and int(n) > 0:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def burbulis_uzlabotais(a):
```

```
    # Burbulis (uzlabotais) kārtošanas metode
```

```
    # Sakārto masīva elementus dilstoša (neaugoša) secībā un
```

izvada paveikto salīdzināšanas skaits, lai sakārtot masīvu. Izmanto burbuļa metodi
(uzlaboto)

```
# a - masīvs
skaititajs = 0
n = len(a)
i = n - 1
paz = True
while paz:
    paz = False
    for j in range(0, i):
        if a[j] < a[j + 1]:
            skaititajs = skaititajs + 1
            paz = True
            x = a[j]
            a[j] = a[j + 1]
            a[j + 1] = x
    i = i - 1
print(skaititajs)
```

```
def izvade(x):
    # Izvada masīva elementus pēc kārtas līdz pedējam
    # x - masīvs
    n = len(x)
    s = str(x[0])
    for i in range(1, n):
        s = s + ", " + str(x[i])
    print(s)
```

```
def izveidot_masivu_ar_garumu(n):
```

```

# Izveido masīvu ar norādīto garumu n

# n - naturāls skaitlis

a = numpy.arange(n)

for i in range(n):

    b = input("Ievadiet " + str(i) + ".elementu ==> ")

    b = is_whole(b, i)

    a[i] = b

return a

```

```

def is_whole(x, i): # Bezgalīgi daudz reizes ievāda

    # Pārbauda vai x ir vesels skaitlis

    # x - pārbaudama simbolu virkne

    # i - elements pēc kārtas

    while True:

        try:

            x = int(x)

        except:

            x = input("Kļūda! Ievadiet " + str(i) + ".elementu ==> ")

        else:

            return int(x)

```

```

# -----

# Galvenā programmas daļa

# -----

```

```

m = input("Ievadiet masīva izmēru N ==> ")

```

```

while is_natural(m) == False:

```

```

m = input("Masīva izmērs ir naturāls skaitlis!\nIevadiet masīva izmēru N ==> ")

m = int(m)

b = izveidot_masivu_ar_garumu(m)

print("Paveikts salīdzināšanas skaits, lai sakārtot masīvu:")

burbulis_uzlabotais(b) # Salīdzināšanas skaits

print("Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:")

izvade(b)

```

Testa piemēri:

1)

```

Ievadiet masīva izmēru N ==> 3
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
3
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
3, 2, 1

```

2)

```

Ievadiet masīva izmēru N ==> -3
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> pieci
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 0
Masīva izmērs ir naturāls skaitlis!
Ievadiet masīva izmēru N ==> 10
Ievadiet 0.elementu ==> 10
Ievadiet 1.elementu ==> 9
Ievadiet 2.elementu ==> 8
Ievadiet 3.elementu ==> 7
Ievadiet 4.elementu ==> 6
Ievadiet 5.elementu ==> 5
Ievadiet 6.elementu ==> 4
Ievadiet 7.elementu ==> 3
Ievadiet 8.elementu ==> 2
Ievadiet 9.elementu ==> 1
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
0
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
10, 9, 8, 7, 6, 5, 4, 3, 2, 1

```

3)

```
Ievadiet masīva izmēru N ==> 10
Ievadiet 0.elementu ==> 1
Ievadiet 1.elementu ==> 2
Ievadiet 2.elementu ==> 3
Ievadiet 3.elementu ==> 4
Ievadiet 4.elementu ==> 5
Ievadiet 5.elementu ==> 6
Ievadiet 6.elementu ==> 7
Ievadiet 7.elementu ==> 8
Ievadiet 8.elementu ==> 9
Ievadiet 9.elementu ==> 10
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
45
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
10, 9, 8, 7, 6, 5, 4, 3, 2, 1
```

4)

```
Ievadiet masīva izmēru N ==> 20
Ievadiet 0.elementu ==> 64
Ievadiet 1.elementu ==> 61
Ievadiet 2.elementu ==> 61
Ievadiet 3.elementu ==> 8
Ievadiet 4.elementu ==> 4
Ievadiet 5.elementu ==> 1
Ievadiet 6.elementu ==> 23
Ievadiet 7.elementu ==> 1896
Ievadiet 8.elementu ==>
Kļūda! Ievadiet 8.elementu ==> 163
Ievadiet 9.elementu ==> 51
Ievadiet 10.elementu ==> 8
Ievadiet 11.elementu ==> 4
Ievadiet 12.elementu ==> 51
Ievadiet 13.elementu ==> 3
Ievadiet 14.elementu ==> 18
Ievadiet 15.elementu ==> 9
Ievadiet 16.elementu ==> 5165
Ievadiet 17.elementu ==> 123
Ievadiet 18.elementu ==> 5
Ievadiet 19.elementu ==> 68
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
92
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
5165, 1896, 163, 123, 68, 64, 61, 61, 51, 51, 23, 18, 9, 8, 8, 5, 4, 4, 3, 1
```


5)

```
Ievadiet masīva izmēru N ==> 5
Ievadiet 0.elementu ==> 491
Ievadiet 1.elementu ==> 6
Ievadiet 2.elementu ==> 49
Ievadiet 3.elementu ==> 6
Ievadiet 4.elementu ==> 12
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
3
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
491, 49, 12, 6, 6
```

6)

```
Ievadiet masīva izmēru N ==> 10
Ievadiet 0.elementu ==> 12.2
Kļūda! Ievadiet 0.elementu ==> 12
Ievadiet 1.elementu ==> 1
Ievadiet 2.elementu ==> 5
Ievadiet 3.elementu ==> 89
Ievadiet 4.elementu ==> 31
Ievadiet 5.elementu ==> 8
Ievadiet 6.elementu ==> 431
Ievadiet 7.elementu ==> 8432
Ievadiet 8.elementu ==> 84
Ievadiet 9.elementu ==> 2
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
28
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
8432, 431, 89, 84, 31, 12, 8, 5, 2, 1
```

7)

```
Ievadiet masīva izmēru N ==> 10
Ievadiet 0.elementu ==> 0
Ievadiet 1.elementu ==> 0
Ievadiet 2.elementu ==> 0
Ievadiet 3.elementu ==> 0
Ievadiet 4.elementu ==> 0
Ievadiet 5.elementu ==> 0
Ievadiet 6.elementu ==> 0
Ievadiet 7.elementu ==> 0
Ievadiet 8.elementu ==> 0
Ievadiet 9.elementu ==> 0
Paveikts salīdzināšanas skaits, lai sakārtot masīvu:
0
Sakārtots skaitļu masīvs dilstoša (neaugoša) secība:
0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```