10. praktiskais darbs. 2. semestris

1. uzdevums

Sastādīt programmu, kas nodrošina determinanta ievadi, aprēķina tā vērtību un izvada uz ekrāna aprēķināto determinanta vērtību.

Kods:

```
# Programmas nosaukums: Determinants

# 1. uzdevums (1MPR10_Vladislavs_Babaņins)

# Uzdevuma formulējums: Sastādit programmu, kas nodrošina determinanta ievadi, aprēķina tā vērtību un izvada uz ekrāna aprēķināto determinanta vērtību.

# Programmas autors: Vladislavs Babaņins

# Versija 1.0



import numpy



def determinants(x):
    # Rekursīvi atrod matricas x determinanta vērtību.
    # Atgriež determinanta vērtību kā skaitļi int no matricas x.
    # x - kvadrātiska n x n divdimensijas numpy masīvs (matrica ar izmēriem n x n)

    n = len(x)
    if n == 1:
        return x[0, 0]
    det = 0
    zime = 1
    for i in range(n):
        xx = numpy.empty((n - 1, n - 1))
        for j in range(1, n):
```

```
            z = 0
            for k in range(n):
                if k != i:
                    xx[j - 1, z] = x[j, k]
                    z = z + 1
        y = determinants(xx)
        det = det + zime * x[0, i] * y
        zime = -zime  # + - + - + ...
    return det


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 ievade_matrica_float(n, m):
    # Lietotājs var ievādīt nXm matricas elementus un funkcija atgriež divdimensijas masīvu ar n rindam un m kolonnam ar ievadītām vērtībam.
    # Ievādītas vērtības ir reālas vērtības (matricas elementi varētu būt float).
    # Glīti izvada atkarīgi no tas, cik ir nepieciešams starpes likt.
    # n - naturāls skaitlis, kurš nosaka matricas rindas skaitu.
    # m - naturāls skaitlis, kurš nosaka matricas kolonnas skaitu.

    a = numpy.empty((n, m), dtype=float)
```

```
        for i in range(n):
            for j in range(m):
                temp = input("Ievadiet matricas elememtu A(" + str(i) + "," + str(j) + ") ===> ")
                while is_real_check(temp) == False:
                    temp = input("Kļūda! Ievadītais elements nav skaitlis!\nIevadiet matricas elememtu A(" + str(i) + "," + str(j) + ") ===> ")

                a[i, j] = float(temp)

        return a


    def is_real_check(n):
        # Pārbauda vai simbolu virkne ir reāls skaitlis vai nav.
        # Atgriež True, ja tas ir reāls skaitlis (float).
        # Atgriež False, ja tas nav reāls skaitlis (float).
        # n - pārbaudāma simbolu virkne.

        try:
            float(n)
        except:
            return False
        else:
            return True


    def matrix_to_string_float(matrix):
        # Atgriež matricas virknes attēlojumu, kur katra rinda ir atdalīta ar \n.
        # Ja vērtība ir vesels skaitlis, tā tiek attēlota bez komata. Pretējā gadījumā tas tiek attēlots ar komatu.
```

```
    # Funkcija atrod arī maksimālo vērtību garumu matricā un papildina nepieciešamus skaitļus ar tujšumiem " ", lai tie būtu pareizi izlīdzināti.

    # matrix - matrica (divdimensijas numpy masīvs ar izmēriem n x m).


    rindas = len(matrix)

    kolonnas = len(matrix[0])

    max_len = 0


    for i in range(rindas):  # atrod max_len, lai uzzinātu cik atkāpes ir nepieciešāmas

        for j in range(kolonnas):

            value = matrix[i][j]

            if value == int(value):

                value_len = len(str(int(value)))

            else:

                value_len = len(str(float(value)))

            if value_len > max_len:

                max_len = value_len


    sv = ""

    # izveido matricas virknes attēlojumu, kur katra rinda ir atdalīta ar \n un izlidzina to pēc skaitļa ar maksimālu garumu

    for i in range(rindas):

        for j in range(kolonnas):

            value = matrix[i][j]

            if value == int(value):

                value = int(value)

            else:

                value = str(float(value))

            atkape = " " * (max_len - len(str(value)))

            sv += atkape + str(value)

            if j < kolonnas - 1:

                sv = sv + " "
```

```
        sv = sv + "\n"

    return sv


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


n = input("Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> ")
while not is_natural(n):
    n = input("Kļūda! Ievadiet determinanta izmēru ==> ")
n = int(n)

a = ievade_matrica_float(n, n)

print("\nJūsu ievadīta matrica A:")
print(matrix_to_string_float(a))

print("det(A) =", determinants(a))
```

Testa piemēri:

1)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 2
Ievadiet matricas elememtu A(0,0) ===> 1
Ievadiet matricas elememtu A(0,1) ===> 2
Ievadiet matricas elememtu A(1,0) ===> 3
Ievadiet matricas elememtu A(1,1) ===> 4

Jūsu ievadīta matrica A:
1 2
3 4

det(A) = -2.0
```

2)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 2
Ievadiet matricas elememtu A(0,0) ===> 0
Ievadiet matricas elememtu A(0,1) ===> 0
Ievadiet matricas elememtu A(1,0) ===> 1
Ievadiet matricas elememtu A(1,1) ===> 1

Jūsu ievadīta matrica A:
0 0
1 1

det(A) = 0.0
```

3)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 3
Ievadiet matricas elememtu A(0,0) ===> 1
Ievadiet matricas elememtu A(0,1) ===> 2
Ievadiet matricas elememtu A(0,2) ===> 3
Ievadiet matricas elememtu A(1,0) ===> 4
Ievadiet matricas elememtu A(1,1) ===> 5
Ievadiet matricas elememtu A(1,2) ===> 6
Ievadiet matricas elememtu A(2,0) ===> 7
Ievadiet matricas elememtu A(2,1) ===> 8
Ievadiet matricas elememtu A(2,2) ===> 9

Jūsu ievadīta matrica A:
1 2 3
4 5 6
7 8 9

det(A) = 0.0
```

4)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 1
Ievadiet matricas elememtu A(0,0) ===> 6

Jūsu ievadīta matrica A:
6

det(A) = 6.0
```

5)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 2
Ievadiet matricas elememtu A(0,0) ===> 1
Ievadiet matricas elememtu A(0,1) ===> 2
Ievadiet matricas elememtu A(1,0) ===> 134513451234
Ievadiet matricas elememtu A(1,1) ===> 3

Jūsu ievadīta matrica A:
           1            2
134513451234            3

det(A) = -269026902465.0
```

6)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 0
Kļūda! Ievadiet determinanta izmēru ==> -1
Kļūda! Ievadiet determinanta izmēru ==> -2
Kļūda! Ievadiet determinanta izmēru ==> 2.3
Kļūda! Ievadiet determinanta izmēru ==> 0.0
Kļūda! Ievadiet determinanta izmēru ==> 0
Kļūda! Ievadiet determinanta izmēru ==> labi
Kļūda! Ievadiet determinanta izmēru ==> pieci
Kļūda! Ievadiet determinanta izmēru ==> 4
Ievadiet matricas elememtu A(0,0) ===> pieci
Kļūda! Ievadītais elements nav skaitlis!
Ievadiet matricas elememtu A(0,0) ===> 0.5
Ievadiet matricas elememtu A(0,1) ===> .
Kļūda! Ievadītais elements nav skaitlis!
Ievadiet matricas elememtu A(0,1) ===>
Kļūda! Ievadītais elements nav skaitlis!
Ievadiet matricas elememtu A(0,1) ===> 5
Ievadiet matricas elememtu A(0,2) ===> 5
Ievadiet matricas elememtu A(0,3) ===> 5
Ievadiet matricas elememtu A(1,0) ===> 6
Ievadiet matricas elememtu A(1,1) ===> 7
Ievadiet matricas elememtu A(1,2) ===> 9
Ievadiet matricas elememtu A(1,3) ===> 12.4444
Ievadiet matricas elememtu A(2,0) ===> 2
Ievadiet matricas elememtu A(2,1) ===> 3
Ievadiet matricas elememtu A(2,2) ===> 5
Ievadiet matricas elememtu A(2,3) ===> 12.3
Ievadiet matricas elememtu A(3,0) ===> 2
Ievadiet matricas elememtu A(3,1) ===> 8
Ievadiet matricas elememtu A(3,2) ===> 4
Ievadiet matricas elememtu A(3,3) ===> 1

Jūsu ievadīta matrica A:
    0.5       5       5       5
      6       7       9 12.4444
      2       3       5    12.3
      2       8       4       1

det(A) = -594.9576
```

7)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 2
Ievadiet matricas elememtu A(0,0) ===> 345123512
Ievadiet matricas elememtu A(0,1) ===> 3422
Ievadiet matricas elememtu A(1,0) ===> 35663246
Ievadiet matricas elememtu A(1,1) ===> -4214

Jūsu ievadīta matrica A:
345123512      3422
 35663246     -4214

det(A) = -1576390107380.0
```

8)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 5
Ievadiet matricas elememtu A(0,0) ===> 2
Ievadiet matricas elememtu A(0,1) ===> 2
Ievadiet matricas elememtu A(0,2) ===> 2
Ievadiet matricas elememtu A(0,3) ===> 2
Ievadiet matricas elememtu A(0,4) ===> 2
Ievadiet matricas elememtu A(1,0) ===> 2
Ievadiet matricas elememtu A(1,1) ===> 2
Ievadiet matricas elememtu A(1,2) ===> 2
Ievadiet matricas elememtu A(1,3) ===> 2
Ievadiet matricas elememtu A(1,4) ===> 2
Ievadiet matricas elememtu A(2,0) ===> 2
Ievadiet matricas elememtu A(2,1) ===> 1.2
Ievadiet matricas elememtu A(2,2) ===> 45234562353542
Ievadiet matricas elememtu A(2,3) ===> 234232
Ievadiet matricas elememtu A(2,4) ===> 342562
Ievadiet matricas elememtu A(3,0) ===> 234
Ievadiet matricas elememtu A(3,1) ===> 78456
Ievadiet matricas elememtu A(3,2) ===> 3
Ievadiet matricas elememtu A(3,3) ===> 2
Ievadiet matricas elememtu A(3,4) ===> 3
Ievadiet matricas elememtu A(4,0) ===> 78
Ievadiet matricas elememtu A(4,1) ===> 4
Ievadiet matricas elememtu A(4,2) ===> 5
Ievadiet matricas elememtu A(4,3) ===> 7
Ievadiet matricas elememtu A(4,4) ===> 4

Jūsu ievadīta matrica A:
             2              2              2              2              2
             2              2              2              2              2
             2            1.2 45234562353542         234232         342562
           234          78456              3              2              3
            78              4              5              7              4

det(A) = 0.0
```

9)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast determinantu ==> 1
Ievadiet matricas elememtu A(0,0) ===> -1245125125324234925924974572 35

Jūsu ievadīta matrica A:
-1245125125324234946696404008 96

det(A) = -1.245125125324235e+29
```

2. uzdevums

Sastādīt programmu, kas nodrošina matricas ievadi, atrod inverso matricu, izmantojot adjunktu metodi, un izvada uz ekrāna gan doto, gan atrast inverso matricu.

Zināšanai: $A^{-1} = \frac{1}{det\ A} \cdot (A^*)^T$, kur A^* ir matricas A adjunktu matrica.

Kods:

```
# Programmas nosaukums:

# 2. uzdevums (1MPR10_Vladislavs_Babaņins)

# Uzdevuma formulējums: Sastādīt programmu, kas nodrošina matricas ievadi, atrod inverso matricu, izmantojot adjunktu metodi un izvada uz ekrāna gan doto,

# gan atrasto inverso matricu.

# Zināšanai: A^-1 = 1/detA * ( A')^T, kur A' ir matricas adjunktu matrica.

# Programmas autors: Vladislavs Babaņins

# Versija 1.0



import numpy



def determinants(x):

    # Rekursīvi atrod matricas x determinanta vērtību.

    # Atgriež determinanta vērtību kā skaitļi no matricas x.

    # x - kvadrātiska n x n divdimensijas numpy masīvs (matrica ar izmēriem n x n)


    n = len(x)

    if n == 1:

        return x[0, 0]

    det = 0

    zime = 1

    for i in range(n):

        xx = numpy.empty((n - 1, n - 1))

        for j in range(1, n):
```

```
            z = 0
            for k in range(n):
                if k != i:
                    xx[j - 1, z] = x[j, k]
                    z = z + 1
        y = determinants(xx)
        det = det + zime * x[0, i] * y
        zime = -zime  # + - + - + ...
    return det


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 ievade_matrica_float(n, m):
    # Lietotājs var ievādīt nXm matricas elementus un funkcija atgriež divdimensijas masīvu ar n rindam un m kolonnam ar ievadītām vērtībam.
    # Ievādītas vērtības ir reālas vērtības (matricas elementi varētu būt float)
    # Glīti izvada atkarīgi no tas, cik ir nepieciešams starpes likt.
    # n - naturāls skaitlis, kurš nosaka matricas rindas skaitu
    # m - naturāls skaitlis, kurš nosaka matricas kolonnas skaitu

    a = numpy.empty((n, m), dtype=float)
```

```
        for i in range(n):
            for j in range(m):
                temp = input("Ievadiet matricas elememtu A(" + str(i) + "," + str(j) + ") ===> ")
                while is_real_check(temp) == False:
                    temp = input("Kļūda! Ievadītais elements nav skaitlis!\nIevadiet matricas elememtu A(" + str(i) + "," + str(j) + ") ===> ")

                a[i, j] = float(temp)

        return a


    def is_real_check(n):
        # Pārbauda vai simbolu virkne ir reāls skaitlis vai nav.
        # Atgriež True, ja tas ir reāls skaitlis (float).
        # Atgriež False, ja tas nav reāls skaitlis (float).
        # n - pārbaudāma simbolu virkne.

        try:
            float(n)
        except:
            return False
        else:
            return True


    def matrix_to_string_float(matrix):
        # Atgriež matricas virknes attēlojumu, kur katra rinda ir atdalīta ar \n.
        # Ja vērtība ir vesels skaitlis, tā tiek attēlota bez komata. Pretējā gadījumā tas tiek attēlots ar komatu.
```

```
    # Funkcija atrod arī maksimālo vērtību garumu matricā un papildina nepieciešamus skaitļus ar tujšumiem " ", lai tie būtu pareizi izlīdzināti.

    # matrix - matrica (divdimensijas numpy masīvs ar izmēriem n x m).


    rindas = len(matrix)

    kolonnas = len(matrix[0])

    max_len = 0


    for i in range(rindas):  # atrod max_len, lai uzzinātu cik atkāpes ir nepieciešāmas

        for j in range(kolonnas):

            value = matrix[i][j]

            if value == int(value):

                value_len = len(str(int(value)))

            else:

                value_len = len(str(float(value)))

            if value_len > max_len:

                max_len = value_len


    sv = ""

    # izveido matricas virknes attēlojumu, kur katra rinda ir atdalīta ar \n un izlidzina to pēc skaitļa ar maksimālu garumu

    for i in range(rindas):

        for j in range(kolonnas):

            value = matrix[i][j]

            if value == int(value):

                value = int(value)

            else:

                value = str(float(value))

            atkape = " " * (max_len - len(str(value)))

            sv += atkape + str(value)

            if j < kolonnas - 1:

                sv = sv + " "
```

```
            sv = sv + "\n"

        return sv


    def transponenta(a):
        # Atgriež transponētu matricu a b[i, j] = a[j, i]
        # a - divdimensijas masīvs (matrica)

        n = a.shape[0]  # x axis
        m = a.shape[1]  # y axis
        b = numpy.empty((m, n))
        for i in range(m):
            for j in range(n):
                b[i, j] = a[j, i]
        return b


    def adjugate_matrix(a):
        # Atgriež dotās kvadrātmatricas a adjunktu matricu a*.
        # a - nXn divdimensijas masīvs (kvadrātmatrica)

        n = a.shape[0]

        if n == 1:
            # Ja matrica is 1x1, tas adjunktu matrica ir tas paša matrica
            return a

        adj = numpy.zeros((n, n))  # Izveidojam tukšo mainīgu ar nullem, kura būs adjunktu matrica
a* matricai a
        for i in range(n):
```

```
            for j in range(n):
                # Izveidojam paligmatricu, noņemot rinda i un kolonnu j no sākotnējās matricas a
                paligmatrica = []
                for k in range(n):
                    if k != i:
                        rinda = []
                        for l in range(n):
                            if l != j:
                                rinda.append(a[k, l])
                        paligmatrica.append(rinda)

                # Konvertējam paligmatricu par NumPy masīvu
                paligmatrica = numpy.array(paligmatrica)
                # Aprēķinam paligmatricas determinantu
                det = determinants(paligmatrica)
                # Reizinām determinantu ar (-1)^(i+j)
                zime = (-1) ** (i + j)
                # Adjunktas matricas elements (i,j) tas ir determinants no palīgmatricas ar zimes reizinājumu (-1 vai +1)
                adj[i, j] = det * zime

        return adj


    def inverse_matrix_by_def(a):
        # Atgriež inverso matrīcu (divdimensijas masīvu) izmantojot transponenta, determinants un adjugate_matrix funckijas
        # a - divdimensijas masīvs (kvadrātiska matrica)

        adj = adjugate_matrix(a)
        transponent_adj_matrix = transponenta(adj)
        det = determinants(a)
```

```
    if det == 0:
        return "Neēksiste"
    else:
        inverse_det = 1 / determinants(a)
        inverse_a = inverse_det * transponent_adj_matrix
        return inverse_a


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


n = input("Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast invērsu matricu ==> ")
while not is_natural(n):
    n = input("Kļūda! Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast invērsu matricu
==> ")
n = int(n)

a = ievade_matrica_float(n, n)

print("\nJūsu ievadīta matrica A:")
print(matrix_to_string_float(a))

print("Atrāsta invērsa matrica A^-1:")
inversa = inverse_matrix_by_def(a)

if str(inversa) == "Neēksiste":
    print("Matricai A neēksistē invērsa matrica A^-1.")
else:
    print(matrix_to_string_float(inversa))
```

Testa piemēri:

1)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast invērsu matricu ==> 3
Ievadiet matricas elememtu A(0,0) ===> 1
Ievadiet matricas elememtu A(0,1) ===> 2
Ievadiet matricas elememtu A(0,2) ===> 3
Ievadiet matricas elememtu A(1,0) ===> 4
Ievadiet matricas elememtu A(1,1) ===> 5
Ievadiet matricas elememtu A(1,2) ===> 6
Ievadiet matricas elememtu A(2,0) ===> 7
Ievadiet matricas elememtu A(2,1) ===> 8
Ievadiet matricas elememtu A(2,2) ===> 9

Jūsu ievadīta matrica A:
1 2 3
4 5 6
7 8 9

Atrāsta invērsa matrica A^-1:
Matricai A neēksistē invērsa matrica A^-1.
```

2)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast invērsu matricu ==> 1
Ievadiet matricas elememtu A(0,0) ===> 1

Jūsu ievadīta matrica A:
1

Atrāsta invērsa matrica A^-1:
1
```

3)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast invērsu matricu ==> 1
Ievadiet matricas elememtu A(0,0) ===> 0

Jūsu ievadīta matrica A:
0

Atrāsta invērsa matrica A^-1:
Matricai A neēksistē invērsa matrica A^-1.
```

4)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast invērsu matricu ==> 2
Ievadiet matricas elememtu A(0,0) ===> 1
Ievadiet matricas elememtu A(0,1) ===> 2
Ievadiet matricas elememtu A(1,0) ===> 3
Ievadiet matricas elememtu A(1,1) ===> 4

Jūsu ievadīta matrica A:
1 2
3 4

Atrāsta invērsa matrica A^-1:
  -2    1
 1.5 -0.5
```

5)

```
Ievadiet kvadrātiskas matricas izmēru, kurai gribat atrast invērsu matricu ==> 4
Ievadiet matricas elememtu A(0,0) ===> 12.2
Ievadiet matricas elememtu A(0,1) ===> 12
Ievadiet matricas elememtu A(0,2) ===> 1
Ievadiet matricas elememtu A(0,3) ===> 4
Ievadiet matricas elememtu A(1,0) ===> 6
Ievadiet matricas elememtu A(1,1) ===> 3
Ievadiet matricas elememtu A(1,2) ===> 9
Ievadiet matricas elememtu A(1,3) ===> 0
Ievadiet matricas elememtu A(2,0) ===> 23
Ievadiet matricas elememtu A(2,1) ===> 24
Ievadiet matricas elememtu A(2,2) ===> 26
Ievadiet matricas elememtu A(2,3) ===> -4.2
Ievadiet matricas elememtu A(3,0) ===> 2
Ievadiet matricas elememtu A(3,1) ===> 3
Ievadiet matricas elememtu A(3,2) ===> 6
Ievadiet matricas elememtu A(3,3) ===> 8

Jūsu ievadīta matrica A:
12.2   12    1    4
   6    3    9    0
  23   24   26 -4.2
   2    3    6    8

Atrāsta invērsa matrica A^-1:
  0.10772958346070675     0.269635315829399   -0.0758579305503663  -0.09369020526929568
 -0.03136215934694252   -0.2827349614483204    0.0851420354833962   0.06038064830225425
-0.061365669191490335   0.02559922104095194   0.0221912752057788  0.042333254078779047
   0.0308526657835445  0.019417365805056454   -0.0296072370730161   0.09402986764489434
```

6)

```
Ievadiet matricas elememtu A(1,2) ===> 7
Ievadiet matricas elememtu A(1,3) ===> 8
Ievadiet matricas elememtu A(1,4) ===> 9
Ievadiet matricas elememtu A(2,0) ===> 0
Ievadiet matricas elememtu A(2,1) ===> 10
Ievadiet matricas elememtu A(2,2) ===> -2
Ievadiet matricas elememtu A(2,3) ===> -3
Ievadiet matricas elememtu A(2,4) ===> -3.5
Ievadiet matricas elememtu A(3,0) ===> 3.5
Ievadiet matricas elememtu A(3,1) ===> 3
Ievadiet matricas elememtu A(3,2) ===> 1
Ievadiet matricas elememtu A(3,3) ===> 2
Ievadiet matricas elememtu A(3,4) ===> 5
Ievadiet matricas elememtu A(4,0) ===> 8
Ievadiet matricas elememtu A(4,1) ===> 11
Ievadiet matricas elememtu A(4,2) ===> 13
Ievadiet matricas elememtu A(4,3) ===> 15
Ievadiet matricas elememtu A(4,4) ===> 16

Jūsu ievadīta matrica A:
   0    0    2    3    4
   5    6    7    8    9
   0   10   -2   -3 -3.5
 3.5    3    1    2    5
   8   11   13   15   16

Atrāsta invērsa matrica A^-1:
 -0.7368421052631579  -1.0961887477313974 -0.16696914700544463  0.36660617059891104   0.6497277676950998
 0.05263157894736842 -0.14337568058076225  0.09074410163339383 0.018148820326678763  0.08166969147005444
   1.368421052631579    7.272232304900181  0.35934664246823955   -1.528130671506352   -3.876588021778584
 -1.8421052631578947   -8.671506352087114  -0.4863883847549909   1.5027223230490017    4.762250453720508
  0.9473684210526315    2.867513611615245   0.1851179673321234  -0.3629764065335753  -1.6333938294010888
```

3. uzdevums

Sastādīt programmu, kas realizē Latloto 5 no 35 izlozi, nodrošina lietotāja vienā loterijas biļetē norādīto piecu skaitļu ievadi un paziņo par laimestu:

- Ja uzminēti 5 skaitļi, tad paziņo “Lielais laimests”
- Ja uzminēti 4 skaitļi, tad paziņo “Vidējais laimests”
- Ja uzminēti 3 skaitļi, tad paziņo “Mazais laimests”
- Ja citādi, tad paziņo “Nav laimesta”

Kods:

Programmas nosaukums: Latloto

3. uzdevums (1MPR10_Vladislavs_Babaņins)

Uzdevuma formulējums: Sastādīt programmu, kas realizē Latloto 5 no 35 izlozi, nodrošina lietot vienā loterijas

biļetē norādīto piecu skaitli ievadi un paziņo par laimestu:

```
# Ja uzminēti 5 skaitli, tad paziņo "Lielais laimests"
# Ja uzminēti 4 skaitļi, tad paziņo "Vidējais laimests"
# Ja uzminēti 3 skaitļi, tad paziņo "Mazais laimests"
# Ja citādi, tad paziņo "Nav laimesta".
# Programmas autors: Vladislavs Babaņins
# Versija 1.0


import random


def random_set_numbers_1_to_35():
    # Atgriež kopu ar nejaušiem skaitļiem kuri neatkartojas no 1 līdz 35. [1, 35].
    # a - set (kopa) - tukša kopa.

    a = set()
    while len(a) < 5:
        b = random.randint(1, 35)
        if b not in a:
            a.add(b)

    return a


def ievadiet_n_skaitlus_seta(n, max_num):
    # Procedura kura ļauj ievādit n skaitļus un ieliekt tos kop (set'ā)
    # n - cik skaitļus ievādīt lietotājam (int)
    # max_num - maksimālais skaitlis līdz kuram tiek veikta loterija (int)

    a = set()  # a - tukša kopa (set)
    for i in range(1, n + 1):
```

```
            while True:
                b = input("Ievadiet " + str(i) + ". skaitli ==> ")
                try:
                    b = int(b)
                    if b < 1 or b > max_num:
                        print("Skaitlim jābūt veselam skaitlim no 1 līdz " + str(max_num) + ".")
                    elif b in a:
                        print("Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!")
                    else:
                        a.add(b)
                        break
                except ValueError:
                    print("Ievadei jābūt skaitlim!")
                    continue

        return a



    def laimests_5_35_izloze(a, b):
        # a - set (kopa) - nejauši izveidota kopa (jāizveido ar funkciju
random_set_numbers_1_to_35(a)).
        # b - set (kopa) - cilvēka izvēlētie skaitļi kopā.

        x = a.intersection(b)
        len1 = len(x)

        if len1 == 5:
            return "Lielais laimests"
        elif len1 == 4:
            return "Vidējais laimests"
        elif len1 == 3:
```

```
            return "Mazais laimests"
        else:
            return "Nav laimesta"


    def cik_atminejat(a, b):
        # Atgriež cik skaitļus Jūs laimējat (atrod kopas šķēlumu)
        # a - set (kopa) - nejauši izveidota kopa (jāizveido ar funkciju
random_set_numbers_1_to_35(a)).
        # b - set (kopa) - cilvēka izvēlētie skaitļi kopā.

        x = a.intersection(b)
        len1 = len(x)

        return len1


    def sort_ievietosana_augosa(a):
        # Sakārto masīvu augoša secība un atgriež salīdzināšanas skaitu, lai sakārtotu masīvu
        # Kārtošanas tiek izmantota ievietošanas metode (insertion sort)
        # a - viendimensijas masīvs

        n = len(a)
        for i in range(1, n):
            if a[i - 1] > a[i]:
                x = a[i]
                j = i
                while a[j - 1] > x:
                    a[j] = a[j - 1]
                    j = j - 1
                    if j == 0:
```

```
                break
            a[j] = x
        return a


    def print_set(a):
        # Izvada (print) uz ekrāna kopu. Bet tas to neatgriež.
        # Jāizsauc vienkarši print_set(a)
        # a - kopa (set)

        elements = sort_ievietosana_augosa(list(a))
        # Iterējam cauri elementu indeksiem un izdrukājiem tos, atdalot tos ar komatiem
        for i in range(len(elements)):
            if i == len(elements) - 1:
                # Ja elements ir pēdējais, izdrukām to bez komata
                print(elements[i])
            else:
                # Ja elements nav pēdējais, izdrukājam to ar komatu. end="", lai nebūtu pārejas uz jauno rindu kā \n
                print(str(elements[i]) + ", ", end="")


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


    a = random_set_numbers_1_to_35()

    print("Sveicināti 'Latloto' 5 no 35 izloze!\n\nIevadiet savus skaitļus no 1 līdz 35.\nJā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'\nJā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'\nJā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'\nCitādi nav laimesta.\n")
```

```
#print("TESTĒŠANAI:", a)  # TESTĒŠANAI (laimēto skaitļu set izvādīt)


b = ievadiet_n_skaitlus_seta(5, 35)
print("\nJūsu izvēlētie skaitļi:")
print_set(b)


print("\nIzlozētie skaitļi:")
print_set(a)
atmineto_skaits = cik_atminejat(a, b)


if atmineto_skaits == 1:  # Pareizam locijumam
   word = "skaitļi"
else:
   word = "skaitļus"


print("\nJūs atminējat " + str(atmineto_skaits) + " " + word + " no 5.")
print("\nJūsu laimests:")
print(laimests_5_35_izloze(a, b))
```

Testa piemēri:

1)

```
Sveicināti 'Latloto' 5 no 35 izloze!

Ievadiet savus skaitļus no 1 līdz 35.
Jā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'
Jā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'
Jā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'
Citādi nav laimesta.

TESTĒŠANAI: {5, 6, 22, 23, 24}
Ievadiet 1. skaitli ==> 1
Ievadiet 2. skaitli ==> 2
Ievadiet 3. skaitli ==> 3
Ievadiet 4. skaitli ==> 4
Ievadiet 5. skaitli ==> 7

Jūsu izvēlētie skaitļi:
1, 2, 3, 4, 7

Izlozētie skaitļi:
5, 6, 22, 23, 24

Jūs atminējat 0 skaitļus no 5.

Jūsu laimests:
Nav laimesta
```

2)

```
Sveicināti 'Latloto' 5 no 35 izloze!

Ievadiet savus skaitļus no 1 līdz 35.
Jā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'
Jā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'
Jā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'
Citādi nav laimesta.

TESTĒŠANAI: {33, 3, 18, 24, 30}
Ievadiet 1. skaitli ==> 33
Ievadiet 2. skaitli ==> 1
Ievadiet 3. skaitli ==> 2
Ievadiet 4. skaitli ==> 4
Ievadiet 5. skaitli ==> 5

Jūsu izvēlētie skaitļi:
1, 2, 4, 5, 33

Izlozētie skaitļi:
3, 18, 24, 30, 33

Jūs atminējat 1 skaitļi no 5.

Jūsu laimests:
Nav laimesta
```

3)

```
Sveicināti 'Latloto' 5 no 35 izloze!

Ievadiet savus skaitļus no 1 līdz 35.
Jā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'
Jā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'
Jā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'
Citādi nav laimesta.

TESTĒŠANAI: {34, 4, 14, 22, 23}
Ievadiet 1. skaitli ==> 34
Ievadiet 2. skaitli ==> 4
Ievadiet 3. skaitli ==> 0
Skaitlim jābūt veselam skaitlim no 1 līdz 35.
Ievadiet 3. skaitli ==> pieci
Ievadei jābūt skaitlim!
Ievadiet 3. skaitli ==> 35
Ievadiet 4. skaitli ==> 34
Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!
Ievadiet 4. skaitli ==> 33
Ievadiet 5. skaitli ==> 32

Jūsu izvēlētie skaitļi:
4, 32, 33, 34, 35

Izlozētie skaitļi:
4, 14, 22, 23, 34

Jūs atminējat 2 skaitļus no 5.

Jūsu laimests:
Nav laimesta
```

4)

```
Sveicināti 'Latloto' 5 no 35 izloze!

Ievadiet savus skaitļus no 1 līdz 35.
Jā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'
Jā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'
Jā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'
Citādi nav laimesta.

TESTĒŠANAI: {33, 3, 7, 10, 23}
Ievadiet 1. skaitli ==> 33
Ievadiet 2. skaitli ==> 3
Ievadiet 3. skaitli ==> 7
Ievadiet 4. skaitli ==> 9
Ievadiet 5. skaitli ==> 22

Jūsu izvēlētie skaitļi:
3, 7, 9, 22, 33

Izlozētie skaitļi:
3, 7, 10, 23, 33

Jūs atminējat 3 skaitļus no 5.

Jūsu laimests:
Mazais laimests
```

5)

```
Sveicināti 'Latloto' 5 no 35 izloze!

Ievadiet savus skaitļus no 1 līdz 35.
Jā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'
Jā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'
Jā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'
Citādi nav laimesta.

TESTĒŠANAI: {3, 4, 16, 17, 20}
Ievadiet 1. skaitli ==> 3
Ievadiet 2. skaitli ==> 4
Ievadiet 3. skaitli ==> 16
Ievadiet 4. skaitli ==> 17
Ievadiet 5. skaitli ==> 1

Jūsu izvēlētie skaitļi:
1, 3, 4, 16, 17

Izlozētie skaitļi:
3, 4, 16, 17, 20

Jūs atminējat 4 skaitļus no 5.

Jūsu laimests:
Vidējais laimests
```

6)

```
Sveicināti 'Latloto' 5 no 35 izloze!

Ievadiet savus skaitļus no 1 līdz 35.
Jā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'
Jā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'
Jā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'
Citādi nav laimesta.

TESTĒŠANAI: {2, 35, 5, 19, 25}
Ievadiet 1. skaitli ==> 2
Ievadiet 2. skaitli ==> 35
Ievadiet 3. skaitli ==> 5
Ievadiet 4. skaitli ==> 19
Ievadiet 5. skaitli ==> 25

Jūsu izvēlētie skaitļi:
2, 5, 19, 25, 35

Izlozētie skaitļi:
2, 5, 19, 25, 35

Jūs atminējat 5 skaitļus no 5.

Jūsu laimests:
Lielais laimests
```

7)

```
Sveicināti 'Latloto' 5 no 35 izloze!

Ievadiet savus skaitļus no 1 līdz 35.
Jā uzminēsiet 5 skaitļus, tad dabūsiet 'Lielo laimestu'
Jā uzminēsiet 4 skaitļus, tad dabūsiet 'Vidējo laimestu'
Jā uzminēsiet 3 skaitļus, tad dabūsiet 'Mazo laimestu'
Citādi nav laimesta.

TESTĒŠANAI: {35, 9, 17, 27, 30}
Ievadiet 1. skaitli ==> 1
Ievadiet 2. skaitli ==> 1
Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!
Ievadiet 2. skaitli ==> 1
Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!
Ievadiet 2. skaitli ==> 1
Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!
Ievadiet 2. skaitli ==> 1
Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!
Ievadiet 2. skaitli ==> 1
Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!
Ievadiet 2. skaitli ==> pieci
Ievadei jābūt skaitlim!
Ievadiet 2. skaitli ==> 12.2
Ievadei jābūt skaitlim!
Ievadiet 2. skaitli ==> 12
Ievadiet 3. skaitli ==> 36
Skaitlim jābūt veselam skaitlim no 1 līdz 35.
Ievadiet 3. skaitli ==> 0
Skaitlim jābūt veselam skaitlim no 1 līdz 35.
Ievadiet 3. skaitli ==> -2
Skaitlim jābūt veselam skaitlim no 1 līdz 35.
Ievadiet 3. skaitli ==> 12.2
Ievadei jābūt skaitlim!
Ievadiet 3. skaitli ==> 12
Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!
Ievadiet 3. skaitli ==> 134
Skaitlim jābūt veselam skaitlim no 1 līdz 35.
Ievadiet 3. skaitli ==> 14
Ievadiet 4. skaitli ==> 15
Ievadiet 5. skaitli ==> 17

Jūsu izvēlētie skaitļi:
1, 12, 14, 15, 17

Izlozētie skaitļi:
9, 17, 27, 30, 35

Jūs atminējat 1 skaitļi no 5.

Jūsu laimests:
Nav laimesta
```

4. uzdevums

Sastādīt programmu, kas realizē kāršu kavas izdali 4 spēlētājiem pa 6 kārtīm katram spēlētājam un paziņo, kādas kārtis katrs spēlētājs ir saņēmis.

Kods:

```
# Programmas nosaukums: Karšu kavas izdali

# 4. uzdevums (1MPR10_Vladislavs_Babaņins)

# Uzdevuma formulējums: Sastādīt programmu, kas realizē kāršu kavas uzdali 4 spēlētājiem pa 6 kārtīm katram spēlētajam

# un paziņo, kādas kārtis katrs spēlētajs ir saņemis.

# Programmas autors: Vladislavs Babaņins

# Versija 1.0


# ♣ ♦ ♥ ♠

# 2 3 4 5 6 7 8 9 10 J Q K A


import random



def karsu_izdale_n_speletajiem_pa_m_kartim(n, m):

    # Atgriež simbolu virknī šāda veidā ar spēlētaju kārtu un viņa kārtem:

    # 1.spēlētājs: ♦6 ♦5 ♠J ♠5 ♠8 ♣A

    # 2.spēlētājs: ♦8 ♠9 ♥6 ♠4 ♣9 ♥Q

    # 3.spēlētājs: ♠7 ♣7 ♠6 ♦10 ♠K ♥4

    # ...

    # n.spēlētajs: ...

    # To vajag print'ēt atsevišķi print(karsu_izdale_n_speletajiem_pa_m_kartim(4, 6))


    # n - int - spēlētaju skaits (naturāls skaitlis)

    # m - int - karšu skaits (naturāls skaitlis)
```

```
    a = set(("♣", "♦", "♥", "♠"))
    b = set(("2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A"))

    c = set()

    for i in a:
        for j in b:
            c.add(i + j)

    res = ""
    for i in range(1, n + 1):
        sv = str(i) + ".spēlētājs:"
        for j in range(m):
            sv = sv + " " + c.pop()
        res = res + sv + "\n"
    return res


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


print("Kāršu kavas izdale 4 spēlētājiem pa 6 kārtīm katram spēlētājam:\n")
print(karsu_izdale_n_speletajiem_pa_m_kartim(4, 6))
```

Testa piemēri:

1)

```
Kāršu kavas izdale 4 spēlētājiem pa 6 kārtīm katram spēlētājam:

1.spēlētājs: ♥J ♣3 ♦4 ♣7 ♥9 ♣6
2.spēlētājs: ♣5 ♠9 ♣4 ♥K ♠6 ♣8
3.spēlētājs: ♣K ♦A ♦6 ♠A ♥A ♦K
4.spēlētājs: ♥Q ♠10 ♥5 ♣A ♦10 ♣2
```

2)

```
Kāršu kavas izdale 4 spēlētājiem pa 6 kārtīm katram spēlētājam:

1.spēlētājs: ♠3 ♣10 ♣6 ♠5 ♥5 ♦5
2.spēlētājs: ♥J ♦K ♠J ♥Q ♠7 ♣A
3.spēlētājs: ♠6 ♦7 ♣3 ♦J ♥7 ♥A
4.spēlētājs: ♣4 ♦4 ♣7 ♥8 ♥2 ♣2
```

3)

```
Kāršu kavas izdale 4 spēlētājiem pa 6 kārtīm katram spēlētājam:

1.spēlētājs: ♦3 ♥4 ♥2 ♣2 ♠9 ♣9
2.spēlētājs: ♣10 ♣5 ♠4 ♣K ♥9 ♥K
3.spēlētājs: ♣Q ♠J ♦10 ♦J ♣6 ♥6
4.spēlētājs: ♦2 ♦7 ♠Q ♥10 ♠A ♠3
```

4)

```
Kāršu kavas izdale 4 spēlētājiem pa 6 kārtīm katram spēlētājam:

1.spēlētājs: ♦4 ♣4 ♥4 ♣Q ♦A ♠K
2.spēlētājs: ♥7 ♥2 ♥K ♠10 ♥9 ♠7
3.spēlētājs: ♦K ♥5 ♠2 ♠Q ♦9 ♥Q
4.spēlētājs: ♣6 ♣A ♦8 ♠J ♣2 ♠4
```

5)

```
Kāršu kavas izdale 4 spēlētājiem pa 6 kārtīm katram spēlētājam:

1.spēlētājs: ♣9 ♣5 ♥Q ♦3 ♠5 ♠6
2.spēlētājs: ♦5 ♠K ♦7 ♥10 ♣7 ♦Q
3.spēlētājs: ♥2 ♠J ♥A ♠10 ♠4 ♠A
4.spēlētājs: ♦9 ♦6 ♥J ♣2 ♦A ♦4
```

6)

```
Kāršu kavas izdale 4 spēlētājiem pa 6 kārtīm katram spēlētājam:

1.spēlētājs: ♥J ♣5 ♠6 ♥4 ♠A ♦A
2.spēlētājs: ♦Q ♥5 ♠9 ♦6 ♦8 ♥K
3.spēlētājs: ♣4 ♠5 ♣3 ♣8 ♦3 ♣2
4.spēlētājs: ♦7 ♠J ♠4 ♣10 ♠10 ♣K
```

5. uzdevums

Sastādīt programmu, kas realizē Keno loterijas izlozi un paziņo laimējošos skaitļus augošā secībā.

Kods:

```
# Programmas nosaukums: Keno loterijas izloze

# 5. uzdevums (1MPR10_Vladislavs_Babaņins)

# Uzdevuma formulējums: Sastādīt programmu, kas realizē Keno loterijas izlozi un paziņo laimējošos skaitļus augošā secība.

# Programmas autors: Vladislavs Babaņins

# Versija 1.0



import random



def keno(n, m):
    # Atgriež kopu ar n-tiem nejaušiem skaitļiem kuri neatkartojas no 1 līdz m. [1, m].
    # a - set (kopa) - tukša kopa.
    # Izvēlas n skaitļus no 1 līdz m, izmantojot sets. Tie neatkartosies
    a = set()
    while len(a) < n:
        b = random.randint(1, m)
        if b not in a:
            a.add(b)

    return a



def sort_ievietosana_augosa(a):
    # Sakārto masīvu augoša secība un atgriež salīdzināšanas skaitu, lai sakārtotu masīvu
    # Kārtošanas tiek izmantota ievietošanas metode (insertion sort)
```

```
    # a - viendimensijas masīvs
    n = len(a)
    for i in range(1, n):
        if a[i - 1] > a[i]:
            x = a[i]
            j = i
            while a[j - 1] > x:
                a[j] = a[j - 1]
                j = j - 1
                if j == 0:
                    break
            a[j] = x
    return a


def print_set(a):
    # Izvada (print) uz ekrāna kopu. Bet tas to neatgriež.
    # Jāizsauc vienkarši print_set(a)
    # a - kopa (set)

    elements = sort_ievietosana_augosa(list(a))
    # Iterējam cauri elementu indeksiem un izdrukājiem tos, atdalot tos ar komatiem
    for i in range(len(elements)):
        if i == len(elements) - 1:
            # Ja elements ir pēdējais, izdrukām to bez komata
            print(elements[i])
        else:
            # Ja elements nav pēdējais, izdrukājam to ar komatu. end="", lai nebūtu pārejas uz jauno rindu kā \n
            print(str(elements[i]) + ", ", end="")
```

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

# Galvenā programmas daļa

# ------------------------------------------------------



# Izvadam uz ekrāna laimējošos skaitļus

print("Keno loterijas izlozes laimesta skaitļi augošā secībā:")

keno = keno(20, 62)

print_set(keno)
```

Testa piemēri:

1)

```
Keno loterijas izlozes laimesta skaitļi augošā secībā:
5, 7, 8, 12, 14, 16, 22, 26, 28, 37, 39, 44, 45, 46, 48, 49, 51, 52, 57, 62
```

2)

```
Keno loterijas izlozes laimesta skaitļi augošā secībā:
2, 3, 6, 8, 12, 13, 20, 25, 26, 28, 34, 35, 36, 38, 42, 43, 44, 46, 53, 62
```

3)

```
Keno loterijas izlozes laimesta skaitļi augošā secībā:
3, 5, 7, 8, 10, 12, 15, 17, 19, 21, 24, 27, 30, 37, 42, 49, 54, 57, 60, 62
```

4)

```
Keno loterijas izlozes laimesta skaitļi augošā secībā:
2, 4, 9, 11, 12, 18, 21, 22, 32, 35, 36, 39, 41, 42, 44, 45, 47, 49, 56, 61
```

5)

```
Keno loterijas izlozes laimesta skaitļi augošā secībā:
1, 3, 6, 9, 16, 17, 20, 25, 26, 30, 31, 33, 36, 37, 44, 54, 55, 56, 59, 60
```

6)

```
Keno loterijas izlozes laimesta skaitļi augošā secībā:
2, 4, 5, 6, 13, 16, 17, 19, 21, 24, 43, 44, 52, 53, 54, 55, 59, 60, 61, 62
```

PU1. uzdevums

Papildināt 5. Uzdevumā realizēto Keno loterijas izlozi ar laimestu sadali.

Kods:

```
# Programmas nosaukums: Keno loterijas izloze ar laimesto sadalījumu

# 5. uzdevums (1MPR10_Vladislavs_Babaņins)

# Uzdevuma formulējums: Papildināt 5.uzdevumā realizēto Keno loterijas izlozi ar laimestu sadali.

# Programmas autors: Vladislavs Babaņins

# Versija 1.0


# Avots: https://www.latloto.lv/lv/keno/

# Noteikumi: https://www.latloto.lv/lv/page/view/2695



import numpy

import random



def is_natural_and_less_than_10(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 and int(n) < 11:

        return True

    else:

        return False



def ievadiet_n_skaitlus_seta(n, max_num):
```

```
    # Procedura kura ļauj ievādit n skaitļus un ieliekt tos kop (set'ā)
    # Atgriež jau aizpildīto kopu (set'u) a ar lietotāja ievādītajiem naturāliem skaitliem robēžas no 1 līdz max_num ieskaitot.
    # n - cik skaitļus ievādīt lietotājam (int)
    # max_num - maksimālais skaitlis līdz kuram tiek veikta loterija (int)

    a = set()
    for i in range(1, n + 1):
        while True:
            b = input("Ievadiet " + str(i) + ". skaitli ==> ")
            try:
                b = int(b)
                if b < 1 or b > max_num:
                    print("Skaitlim jābūt veselam skaitlim no 1 līdz " + str(max_num) + ".")
                elif b in a:
                    print("Šis skaitlis jau eksistē, lūdzu ievadiet citu skaitli!")
                else:
                    a.add(b)
                    break
            except ValueError:
                print("Ievadei jābūt skaitlim!")
                continue

    return a


def speles_likmes():
    # Paprasam izvēlēties spēlēs likmi no piedāvātiem.
    # Pārbauda to, vai tāda eksistē un atgriež to, kā float.

    while True:
```

```
n = input("Izvēlēties, Jūsu spēles likmi € ==> ")
try:
    float(n)
except:
    print("Kļūda! Ievādiet reālo likmi no piedāvatiem!")
else:
    n = float(n)
    if n == 0.2:
        likme = 0.2
        return likme

    elif n == 0.3:
        likme = 0.3
        return likme

    elif n == 0.5:
        likme = 0.5
        return likme

    elif n == 1:
        likme = 1
        return likme

    elif n == 2:
        likme = 2
        return likme

    elif n == 3:
        likme = 3
        return likme
```

```
        elif n == 5:
            likme = 5
            return likme

        elif n == 10:
            likme = 10
            return likme

        else:
            print("Kļūda! Ievādiet likmi no piedāvatiem!")


def laimestu_matrica(izveleto_skaits, atmineto_skaits):
    # Atgriež reizinātāju pamatojoties uz laimestu sadalījuma matricas.
    # Reizinātajs ir atkarīgs no tā, cik skaitļus jus izvēlējaties un cik jus atminējat.
    # Reizinātāji ir pārādīta šājā matricā.

    laimesti = numpy.array([[0, 0, 0, 1, 1, 1, 1, 1, 1, 1],
                            [1.5, 0, 0, 0, 0, 0, 0, 0, 0, 0],
                            [0, 4.5, 1, 0, 0, 0, 0, 0, 0, 0],
                            [0, 0, 8, 1, 1, 0, 0, 0, 0, 0],
                            [0, 0, 0, 20, 2, 2, 1, 0, 0, 0],
                            [0, 0, 0, 0, 45, 12, 3, 3, 1, 1],
                            [0, 0, 0, 0, 0, 175, 30, 5, 2, 2],
                            [0, 0, 0, 0, 0, 0, 700, 100, 40, 5],
                            [0, 0, 0, 0, 0, 0, 0, 3000, 350, 55],
                            [0, 0, 0, 0, 0, 0, 0, 0, 10000, 550],
                            [0, 0, 0, 0, 0, 0, 0, 0, 0, 60000]])

        reizinatajs = laimesti[atmineto_skaits, izveleto_skaits - 1]  # Izvēlāmies nepieciešāmo rūtiņu
un tas arī būs reizinātajs.
```

```
    return reizinatajs



def naudas_balva(likme, reizinatajs):
    # Atgriež naudas balvu likmi reizinot ar reizinātaju.
    # likme - likme (float reāls skaitlis lielāks par 0, piemēram,
    # 0.20
    # 0.30
    # 0.50
    # 1.00
    # 2.00
    # 3.00
    # 5.00
    # 10.00)
    # reizinātajs - float reāls skaitlis lielāks vai vienāds par 0

    return likme * reizinatajs



def random_set_numbers_skaita_n_from_1_to_m(n, m):
    # Atgriež kopu ar n nejaušiem skaitļiem, kuri neatkartojas no 1 līdz m. [1, m].
    # n - cik nejaušus skaitļus vajag loterijai
    # m - kāda ir augšēja robeža skaitlim, kuru vajadzīgi uzminēt (vislielākais iespējamais skaitlis loterija)

    a = set()
    while len(a) < n:
        b = random.randint(1, m)
        if b not in a:
            a.add(b)
```

```
    return a


def cik_atminejat(a, b):
    # Atgriež cik skaitļus Jūs laimējat (atrod kopas šķēlumu)
    # a - set (kopa) - nejauši izveidota kopa (jāizveido ar funkciju random_set_numbers_1_to_35(a)).
    # b - set (kopa) - cilvēka izvēlētie skaitļi kopā.

    x = a.intersection(b)
    len1 = len(x)

    return len1


def print_set(a):
    # Izvada (print) uz ekrāna kopu. Bet tas to neatgriež.
    # Jāizsauc vienkarši print_set(a)
    # a - kopa (set)

    elements = sort_ievietosana_augosa(list(a))
    # Iterējam cauri elementu indeksiem un izdrukājiem tos, atdalot tos ar komatiem
    for i in range(len(elements)):
        if i == len(elements) - 1:
            # Ja elements ir pēdējais, izdrukām to bez komata
            print(elements[i])
        else:
            # Ja elements nav pēdējais, izdrukājam to ar komatu. end="", lai nebūtu pārejas uz jauno rindu kā \n
            print(str(elements[i]) + ", ", end="")
```

```
def sort_ievietosana_augosa(a):
    # Sakārto masīvu augoša secība un atgriež salīdzināšanas skaitu, lai sakārtotu masīvu
    # Kārtošanas tiek izmantota ievietošanas metode (insertion sort)
    # a - viendimensijas masīvs

    n = len(a)
    for i in range(1, n):
        if a[i - 1] > a[i]:
            x = a[i]
            j = i
            while a[j - 1] > x:
                a[j] = a[j - 1]
                j = j - 1
                if j == 0:
                    break
            a[j] = x
    return a


def izlozu_skaits():
    # Paprasam izvēlēties izložu skaitu no piedāvātiem.
    # Pārbauda to, vai tāda eksistē un atgriež to, kā float.

    while True:
        n = input("Izvēlēties, izložu skaitu ==> ")
        try:
            int(n)
        except:
            print("Kļūda! Ievādiet reālo izložu skaitu no piedāvatiem!")
        else:
```

```
n = int(n)
if n == 1:
    izlozu_skaits = 1
    return izlozu_skaits

elif n == 2:
    izlozu_skaits = 2
    return izlozu_skaits

elif n == 3:
    izlozu_skaits = 3
    return izlozu_skaits

elif n == 4:
    izlozu_skaits = 4
    return izlozu_skaits

elif n == 6:
    izlozu_skaits = 6
    return izlozu_skaits

elif n == 12:
    izlozu_skaits = 12
    return izlozu_skaits

elif n == 21:
    izlozu_skaits = 21
    return izlozu_skaits

else:
    print("Kļūda! Ievādiet izložu skaitu no piedāvatiem!")
```

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


n = input("Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> ")
while not is_natural_and_less_than_10(n):
    n = input("Kļūda! Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> ")
print("Ievādiet Jūsu skaitļus. Skaitlim jābūt veselam skaitlim no 1 līdz 62.")
n = int(n)

rezultats = random_set_numbers_skaita_n_from_1_to_m(20, 62)
# print("TESTĒŠANAI:")  # TESTĒŠANAI
# print_set(rezultats)  # TESTĒŠANAI

players_numbers = ievadiet_n_skaitlus_seta(n, 62)

print("\nSpēles likmes:\n0.20 €\n0.30 €\n0.50 €\n1.00 €\n2.00 €\n3.00 €\n5.00 €\n10.00 €\n")
likme = speles_likmes()

print("\nIzložu skaits:\n1\n2\n3\n4\n6\n12\n21\n")
piedalisanas_reizes = izlozu_skaits()

print("\nJūsu skaitļi:")
print_set(players_numbers)

total_laimests = 0
```

```
for i in range(piedalisanas_reizes):

    rezultats = random_set_numbers_skaita_n_from_1_to_m(20, 62)  # Noņemt testēšanai
    print("\nIzlozētie skaitļi:")
    print_set(rezultats)

    atmineto_skaits = cik_atminejat(players_numbers, rezultats)

    if atmineto_skaits == 1:  # Pareizam locijumam
        word = " skaitļi."
    else:
        word = " skaitļus."

    print("\nJūs atminējat " + str(atmineto_skaits) + word)
    reizinatajs = laimestu_matrica(n, atmineto_skaits)
    print("Reizinātājs:", reizinatajs)

    laimests = naudas_balva(likme, reizinatajs)
    print("Jūsu laimests:", laimests, "€")
    total_laimests += laimests

    if piedalisanas_reizes == 1:
        pass
    elif i + 1 == piedalisanas_reizes:
        if piedalisanas_reizes == 21:  # Pareizam locijumam
            word = "izlozi"
        else:
            word = "izlozem"
        print("\nJūsu kopējais laimests par", piedalisanas_reizes, word, "ir:", total_laimests, "€")
```

Testa piemēri:

1)

```
Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> 1
Ievādiet Jūsu skaitļus. Skaitlim jābūt veselam skaitlim no 1 līdz 62.
Ievadiet 1. skaitli ==> 1

Spēles likmes:
0.20 €
0.30 €
0.50 €
1.00 €
2.00 €
3.00 €
5.00 €
10.00 €

Izvēlēties, Jūsu spēles likmi € ==> 10

Izložu skaits:
1
2
3
4
6
12
21

Izvēlēties, izložu skaitu ==> 1

Jūsu skaitļi:
1

Izlozētie skaitļi:
5, 7, 10, 18, 21, 28, 32, 33, 37, 38, 40, 41, 42, 43, 44, 46, 47, 49, 51, 55

Jūs atminējat 0 skaitļus.
Reizinātājs: 0.0
Jūsu laimests: 0.0 €
```

2)

```
Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> 2
Ievādiet Jūsu skaitļus. Skaitlim jābūt veselam skaitlim no 1 līdz 62.
Ievadiet 1. skaitli ==> 2
Ievadiet 2. skaitli ==> 3

Spēles likmes:
0.20 €
0.30 €
0.50 €
1.00 €
2.00 €
3.00 €
5.00 €
10.00 €

Izvēlēties, Jūsu spēles likmi € ==> 3

Izložu skaits:
1
2
3
4
6
12
21

Izvēlēties, izložu skaitu ==> 4

Jūsu skaitļi:
2, 3

Izlozētie skaitļi:
1, 4, 7, 10, 14, 16, 23, 24, 28, 31, 33, 34, 35, 36, 48, 49, 52, 53, 54, 55

Jūs atminējat 0 skaitļus.
Reizinātājs: 0.0
Jūsu laimests: 0.0 €

Izlozētie skaitļi:
1, 2, 3, 5, 8, 11, 12, 17, 20, 22, 23, 24, 25, 28, 29, 33, 49, 51, 53, 59

Jūs atminējat 2 skaitļus.
Reizinātājs: 4.5
Jūsu laimests: 13.5 €

Izlozētie skaitļi:
1, 2, 3, 8, 12, 13, 14, 18, 20, 27, 31, 33, 34, 35, 44, 47, 53, 56, 59, 61

Jūs atminējat 2 skaitļus.
Reizinātājs: 4.5
Jūsu laimests: 13.5 €

Izlozētie skaitļi:
6, 17, 19, 26, 27, 29, 32, 33, 35, 37, 38, 39, 40, 43, 46, 50, 52, 55, 57, 59

Jūs atminējat 0 skaitļus.
Reizinātājs: 0.0
Jūsu laimests: 0.0 €

Jūsu kopējais laimests par 4 izlozem ir: 27.0 €
```

3)

```
Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> 1
Ievādiet Jūsu skaitļus. Skaitlim jābūt veselam skaitlim no 1 līdz 62.
Ievadiet 1. skaitli ==> 62

Spēles likmes:
0.20 €
0.30 €
0.50 €
1.00 €
2.00 €
3.00 €
5.00 €
10.00 €

Izvēlēties, Jūsu spēles likmi € ==> 10

Izložu skaits:
1
2
3
4
6
12
21

Izvēlēties, izložu skaitu ==> 2

Jūsu skaitļi:
62

Izlozētie skaitļi:
3, 4, 10, 11, 12, 16, 18, 19, 24, 28, 29, 33, 39, 44, 45, 49, 50, 54, 59, 62

Jūs atminējat 1 skaitļi.
Reizinātājs: 1.5
Jūsu laimests: 15.0 €

Izlozētie skaitļi:
1, 4, 11, 12, 13, 21, 25, 29, 33, 34, 36, 45, 47, 49, 50, 53, 55, 57, 58, 62

Jūs atminējat 1 skaitļi.
Reizinātājs: 1.5
Jūsu laimests: 15.0 €

Jūsu kopējais laimests par 2 izlozem ir: 30.0 €
```

4)

```
Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> 10
Ievādiet Jūsu skaitļus. Skaitlim jābūt veselam skaitlim no 1 līdz 62.
Ievadiet 1. skaitli ==> 1
Ievadiet 2. skaitli ==> 2
Ievadiet 3. skaitli ==> 3
Ievadiet 4. skaitli ==> 4
Ievadiet 5. skaitli ==> 5
Ievadiet 6. skaitli ==> 6
Ievadiet 7. skaitli ==> 7
Ievadiet 8. skaitli ==> 8
Ievadiet 9. skaitli ==> 12.2
Ievadei jābūt skaitlim!
Ievadiet 9. skaitli ==> pieci
Ievadei jābūt skaitlim!
Ievadiet 9. skaitli ==> 12
Ievadiet 10. skaitli ==> 13

Spēles likmes:
0.20 €
0.30 €
0.50 €
1.00 €
2.00 €
3.00 €
5.00 €
10.00 €

Izvēlēties, Jūsu spēles likmi € ==> 12
Kļūda! Ievādiet likmi no piedāvatiem!
Izvēlēties, Jūsu spēles likmi € ==> pieci
Kļūda! Ievādiet reālo likmi no piedāvatiem!
Izvēlēties, Jūsu spēles likmi € ==> 3

Izložu skaits:
1
2
3
4
6
12
21

Izvēlēties, izložu skaitu ==> 10
Kļūda! Ievādiet izložu skaitu no piedāvatiem!
Izvēlēties, izložu skaitu ==> 1

Jūsu skaitļi:
1, 2, 3, 4, 5, 6, 7, 8, 12, 13

Izlozētie skaitļi:
1, 4, 5, 6, 7, 10, 17, 20, 23, 34, 36, 44, 46, 47, 50, 51, 52, 55, 57, 58

Jūs atminējat 5 skaitļus.
Reizinātājs: 1.0
Jūsu laimests: 3.0 €
```

5)

```
Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> 100
Kļūda! Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> labi
Kļūda! Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> 6
Ievādiet Jūsu skaitļus. Skaitlim jābūt veselam skaitlim no 1 līdz 62.
Ievadiet 1. skaitli ==> 1
Ievadiet 2. skaitli ==> 2
Ievadiet 3. skaitli ==> 3
Ievadiet 4. skaitli ==> 63
Skaitlim jābūt veselam skaitlim no 1 līdz 62.
Ievadiet 4. skaitli ==> 30
Ievadiet 5. skaitli ==> 32
Ievadiet 6. skaitli ==> 33

Spēles likmes:
0.20 €
0.30 €
0.50 €
1.00 €
2.00 €
3.00 €
5.00 €
10.00 €

Izvēlēties, Jūsu spēles likmi € ==> 2

Izložu skaits:
1
2
3
4
6
12
21

Izvēlēties, izložu skaitu ==> 6

Jūsu skaitļi:
1, 2, 3, 30, 32, 33

Izlozētie skaitļi:
1, 2, 4, 7, 10, 11, 13, 14, 17, 20, 21, 22, 32, 33, 34, 37, 46, 53, 57, 60

Jūs atminējat 4 skaitļus.
Reizinātājs: 2.0
Jūsu laimests: 4.0 €

Izlozētie skaitļi:
6, 8, 9, 16, 17, 20, 23, 24, 28, 31, 33, 37, 42, 50, 51, 52, 54, 55, 56, 58

Jūs atminējat 1 skaitļi.
Reizinātājs: 0.0
Jūsu laimests: 0.0 €

Izlozētie skaitļi:
5, 7, 8, 11, 12, 15, 20, 23, 24, 25, 31, 34, 38, 39, 46, 47, 55, 58, 60, 61

Jūs atminējat 0 skaitļus.
Reizinātājs: 1.0
Jūsu laimests: 2.0 €

Izlozētie skaitļi:
8, 9, 10, 16, 18, 27, 28, 31, 38, 39, 41, 46, 47, 49, 50, 51, 55, 57, 58, 62

Jūs atminējat 0 skaitļus.
Reizinātājs: 1.0
Jūsu laimests: 2.0 €

Izlozētie skaitļi:
1, 3, 6, 14, 16, 17, 18, 26, 27, 30, 33, 36, 41, 46, 49, 53, 56, 60, 61, 62

Jūs atminējat 4 skaitļus.
Reizinātājs: 2.0
Jūsu laimests: 4.0 €

Izlozētie skaitļi:
5, 7, 11, 12, 16, 21, 22, 24, 27, 28, 29, 32, 35, 37, 38, 43, 50, 51, 54, 59

Jūs atminējat 1 skaitļi.
Reizinātājs: 0.0
Jūsu laimests: 0.0 €

Jūsu kopējais laimests par 6 izlozem ir: 12.0 €
```

6)

```
Izvēlēties, cik skaitļus nosvītrot (no 1 līdz 10) ==> 10
Ievādiet Jūsu skaitļus. Skaitlim jābūt veselam skaitlim no 1 līdz 62.
TESTĒŠANAI:
3, 5, 6, 8, 10, 21, 22, 24, 25, 26, 30, 32, 36, 39, 43, 47, 50, 51, 55, 60
Ievadiet 1. skaitli ==> 3
Ievadiet 2. skaitli ==> 5
Ievadiet 3. skaitli ==> 6
Ievadiet 4. skaitli ==> 8
Ievadiet 5. skaitli ==> 10
Ievadiet 6. skaitli ==> 21
Ievadiet 7. skaitli ==> 22
Ievadiet 8. skaitli ==> 24
Ievadiet 9. skaitli ==> 25
Ievadiet 10. skaitli ==> 26

Spēles likmes:
0.20 €
0.30 €
0.50 €
1.00 €
2.00 €
3.00 €
5.00 €
10.00 €

Izvēlēties, Jūsu spēles likmi € ==> 10

Izložu skaits:
1
2
3
4
6
12
21

Izvēlēties, izložu skaitu ==> 1

Jūsu skaitļi:
3, 5, 6, 8, 10, 21, 22, 24, 25, 26

Izlozētie skaitļi:
3, 5, 6, 8, 10, 21, 22, 24, 25, 26, 30, 32, 36, 39, 43, 47, 50, 51, 55, 60

Jūs atminējat 10 skaitļus.
Reizinātājs: 60000.0
Jūsu laimests: 600000.0 €
```