

1MPR09_Simona_Bļinova sb24037

1.uzdevums

Programma, kas reālizē biļešu iegādi.

Kods:

```
import numpy
import math
import random
```

```
def zale(a, b):
    c = numpy.empty((a, b+1))
    for i in range(a):
        c[i,0] = i+1
        for j in range(b):
            c[i, j+1] = j+1
    c = numpy.array(c, dtype='i')
    return c
```

```
def aiznemts(a):
    n = a.shape[0]
    m = a.shape[1]
    vs = n * (m-1)
    avs = math.floor(vs * 0.5)
    #print(avs)
    while avs >= 0:
        randn = random.randint(1, n)
        randm = random.randint(1, m-1)
        a[randn-1, randm] = 0
        avs -= 1
    return a
```

```
def izvade(a):
    n = a.shape[0]
    m = a.shape[1]

    if len(str(n)) >= len(str(m-1)):
        garums = len(str(n))
    else:
```

```
garums = len(str(m-1))
```

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

```
    rinda = "
```

```
    for j in range(m):
```

```
        b = str(a[i, j])
```

```
        c = garums - len(b)
```

```
        if c != 0:
```

```
            rinda += ' '*c
```

```
        if b == '0':
```

```
            rinda += 'X'
```

```
        else:
```

```
            rinda += b
```

```
        if j == 0:
```

```
            rinda += '| '
```

```
        else:
```

```
            rinda += ' '
```

```
print(rinda)
```

```
rindas = int(input('Ievadiet rindu skaitu teātrī --> '))
```

```
vietas = int(input('Ievadiet vietu skaitu rindā --> '))
```

```
print(' ')
```

```
zale = zale(rindas, vietas)
```

```
zale = aiznemts(zale)
```

```
#izvade(zale)
```

```
print('Sedvietu iegāde')
```

```
print(' ')
```

```
izpardots = numpy.empty((rindas, vietas+1))
```

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

```
    izpardots[i, 0] = i+1
```

```
    for j in range(vietas):
```

```
        izpardots[i, j+1] = 0
```

```
izpardots = numpy.array(izpardots, dtype='i')
```

```
#izvade(izpardots)
```

```
paz = True
```

```
while paz:
```

```
    if (zale == izpardots).all():
```

```
        print('Biļetes ir izpārdotas!')
```

```
        paz = False
```

```
    else:
```

```
        rinda = int(input('Ievadiet rindu --> '))
```

```
        vieta = int(input('Ievadiet vietas numuru --> '))
```

```
        if rinda > zale.shape[0] or vieta > zale.shape[1]-1:
```

```
            print('Tādas sedvietas nav!')
```

```
            izvade(zale)
```

```
        else:
```

```
            if rinda == 0 and vieta == 0:
```

```
                paz = False
```

```
            else:
```

```
                if zale[rinda-1, vieta] == 0:
```

```
                    print('Šī vieta ir aizņemta!')
```

```
                    izvade(zale)
```

```
                else:
```

```
                    zale[rinda-1, vieta] = 0
```

```
#izvade(zale)
```

```
Testa piemērs(1)
```

```
Ievadiet rindu skaitu teātrī --> 3
Ievadiet vietu skaitu rindā --> 4

Sedvietu iegāde

Ievadiet rindu --> 2
Ievadiet vietas numuru --> 3
Ievadiet rindu --> 1
Ievadiet vietas numuru --> 1
Šī vieta ir aizņemta!
1| X X 3 X
2| X X X 4
3| 1 X 3 4
Ievadiet rindu --> 1
Ievadiet vietas numuru --> 3
Ievadiet rindu --> 0
Ievadiet vietas numuru --> 0
```

Testa piemērs(2)

```
Ievadiet rindu skaitu teātri --> 12
Ievadiet vietu skaitu rindā --> 12

Sedvietu iegāde

Ievadiet rindu --> 12
Ievadiet vietas numuru --> 5
Šī vieta ir aizņemta!
1| X 2 3 X X X 7 8 9 X 11 12
2| 1 X 3 4 5 X 7 8 9 10 X 12
3| X X 3 4 X 6 7 8 X X X 12
4| X 2 X X 5 6 7 8 9 10 11 12
5| X 2 3 4 X X 7 X 9 X X X
6| 1 X X X 5 6 7 8 X 10 11 12
7| X X X X 5 X 7 8 9 10 11 12
8| 1 X X 4 5 X X X X 10 11 12
9| 1 2 X X X 6 7 8 9 X 11 12
10| X 2 3 4 5 6 X 8 X X 11 12
11| 1 X 3 4 5 X X 8 X X 11 X
12| 1 X 3 4 X X X 8 X 10 11 X
Ievadiet rindu --> 6
Ievadiet vietas numuru --> 6
Ievadiet rindu --> 0
Ievadiet vietas numuru --> 0
```

Testa piemērs(3)

```
Ievadiet rindu skaitu teātri --> 3
Ievadiet vietu skaitu rindā --> 10

Sedvietu iegāde

Ievadiet rindu --> 10
Ievadiet vietas numuru --> 2
Tādas sedvietas nav!
1| 1 X X X X 6 7 8 X X
2| 1 X 3 4 5 6 7 X 9 X
3| X X 3 4 5 6 X 8 9 X
Ievadiet rindu --> 2
Ievadiet vietas numuru --> 3
Ievadiet rindu --> 2
Ievadiet vietas numuru --> 4
Ievadiet rindu --> 2
Ievadiet vietas numuru --> 5
Ievadiet rindu --> 0
Ievadiet vietas numuru --> 0
```

2.uzdevums

Programma, kas veic labirinta izveidi ar šķēršļiem un pārbauda vai to iespējams iziet.

Kods:

```
import numpy
import random
import math

def parbaude(a, b, c):
    try:
        a = int(a)
```

```

    if a < 1 or a > b*c:
        raise Exception
    else:
        return int(a)
except:
    print('Nepareiza vērtība. Programma beidz darbību!')
    exit()

```

```

def labirinta_izveide1(a, b):
    c = numpy.empty((a, b))
    for i in range(a):
        for j in range(b):
            sk = input('Ievadiet labirinta '+str(i+1)+'.'rindas '+str(j+1)+'.'kolonnas skaitli --> ')
            sk = parbaude(sk, a, b)
            c[i, j] = sk
    c = numpy.array(c, dtype='i')
    return c

```

```

def labirinta_izveide2(a, b):
    c = numpy.empty((a, b))
    for i in range(a):
        for j in range(b):
            c[i, j] = random.randint(1, a*b)
    c = numpy.array(c, dtype='i')
    return c

```

```

def skersli(a):
    n = a.shape[0]
    m = a.shape[1]
    vs = n * m
    avs = round(vs * 0.05)
    #print(avs)
    while avs > 0:
        randn = random.randint(1, n)
        randm = random.randint(1, m-1)
        a[randn-1, randm] = 0
        avs -= 1
    return a

```

```

def cels(a, b, l, m):
    n1 = l.shape[0]-1
    n2 = l.shape[1]-1
    if a == n1 and b == n2:
        return True
    else:
        ircels = False
        if a < n1 and l[a+1, b] >= l[a, b]:
            m[a+b] = 'uz leju'
            ircels = cels(a+1, b, l, m)
        if not ircels and b < n2 and l[a, b+1] >= l[a, b]:
            m[a+b] = 'pa labi'
            ircels = cels(a, b+1, l, m)
        return ircels

```

```

def izvade(a):
    n = a.shape[0]
    m = a.shape[1]

    garums = len(str(n*m))

```

```

    for i in range(n):
        rinda = ""
        for j in range(m):
            b = str(a[i, j])
            c = garums - len(b)

```

```

            if c != 0:
                rinda += ' '*c

```

```

            if b == '0':
                rinda += 'X'

```

```

            else:
                rinda += b

```

```

        rinda += ' '

```

```

    print(rinda)

```

```
n = int(input('Ievadiet labirinta rindu skaitu --> '))
m = int(input('Ievadiet labirinta kolonnu skaitu --> '))
```

```
labirints = labirinta_izveide1(n, m)
labirints = skersli(labirints)
izvade(labirints)
```

```
garums_cels = n + m - 2
marsruts = numpy.empty(garums_cels, 'O')
```

```
if cels(0, 0, labirints, marsruts):
    print('Labirintu var iziet virzoties', marsruts)
else:
    print('Labirints nav izejams!')
```

Testa piemērs(1)

```
Ievadiet labirinta rindu skaitu --> 3
Ievadiet labirinta kolonnu skaitu --> 3
Ievadiet labirinta 1.rindas 1.kolonnas skaitli --> 1
Ievadiet labirinta 1.rindas 2.kolonnas skaitli --> 2
Ievadiet labirinta 1.rindas 3.kolonnas skaitli --> 3
Ievadiet labirinta 2.rindas 1.kolonnas skaitli --> 3
Ievadiet labirinta 2.rindas 2.kolonnas skaitli --> 4
Ievadiet labirinta 2.rindas 3.kolonnas skaitli --> 5
Ievadiet labirinta 3.rindas 1.kolonnas skaitli --> 1
Ievadiet labirinta 3.rindas 2.kolonnas skaitli --> 5
Ievadiet labirinta 3.rindas 3.kolonnas skaitli --> 6
1 2 3
3 4 5
1 5 6
Labirintu var iziet virzoties ['uz leju' 'pa labi' 'uz leju' 'pa labi']
```

Testa piemērs(2)

```
1 3 X 2 4 5
8 2 3 7 8 9
4 4 6 X 3 9
5 3 7 8 9 10
7 8 12 13 13 16
Labirints nav izejams!
```

Testa piemērs(3)

```
1 2 5
3 4 6
4 8 8
6 7 X
Labirints nav izejams!
```

3.uzdevums

Programma, kas aprēķina matricas determinantu.

Kods:

```
import numpy
```

```
def izveidot_matricu(a, b):  
    c = numpy.empty((a, b))  
    for i in range(a):  
        for j in range(b):  
            c[i, j] = int(input('Ievadiet matricas '+str(i+1)+'-rindas '+str(j+1)+'-kolonnas skaitli --> '))  
    return c
```

```
def determinants(a):  
    n = a.shape[0]  
    n1 = a.shape[1]  
    if n != n1:  
        return 'Šī nav kvadrātiskā matrica, nevar aprēķināt determinantu.'  
    det = 1  
    for u in range(n):  
        if a[u, u] == 0:  
            k = u  
            while a[k, u] == 0:  
                k += 1  
            if k >= n:  
                return 0  
            det = -det  
            for i in range(n):  
                x = a[u, i]  
                a[u, i] = a[k, i]  
                a[k, i] = x  
            det = det * a[u, u]  
            for j in range(n-1, u-1, -1):  
                a[u, j] = a[u, j] / a[u, u]  
            for i in range(u+1, n):  
                for j in range(n-1, u-1, -1):  
                    a[i, j] = a[i, j] - a[i, u] * a[u, j]  
    return f'det = {det}'
```

```
n = int(input('Ievadiet matricas rindu skaitu --> '))  
m = int(input('Ievadiet matricas kolonnu skaitu --> '))
```

```
matrica = izveidot_matricu(n, m)
```



```
print(determinants(matrica))
```

Testa piemērs(1)

```
Ievadiet matricas rindu skaitu --> 2
Ievadiet matricas kolonnu skaitu --> 2
Ievadiet matricas 1.rindas 1.kolonnas skaitli --> 1
Ievadiet matricas 1.rindas 2.kolonnas skaitli --> 2
Ievadiet matricas 2.rindas 1.kolonnas skaitli --> 2
Ievadiet matricas 2.rindas 2.kolonnas skaitli --> 1
det = -3.0
```

Testa piemērs(2)

```
Ievadiet matricas rindu skaitu --> 2
Ievadiet matricas kolonnu skaitu --> 3
Ievadiet matricas 1.rindas 1.kolonnas skaitli --> 3
Ievadiet matricas 1.rindas 2.kolonnas skaitli --> -6
Ievadiet matricas 1.rindas 3.kolonnas skaitli --> 2
Ievadiet matricas 2.rindas 1.kolonnas skaitli --> 4
Ievadiet matricas 2.rindas 2.kolonnas skaitli --> 12
Ievadiet matricas 2.rindas 3.kolonnas skaitli --> 5
ši nav kvadrātiskā matrica, nevar aprēķināt determinantu.
```

Testa piemērs(3)

```
Ievadiet matricas rindu skaitu --> 3
Ievadiet matricas kolonnu skaitu --> 3
Ievadiet matricas 1.rindas 1.kolonnas skaitli --> 1
Ievadiet matricas 1.rindas 2.kolonnas skaitli --> 2
Ievadiet matricas 1.rindas 3.kolonnas skaitli --> 1
Ievadiet matricas 2.rindas 1.kolonnas skaitli --> 4
Ievadiet matricas 2.rindas 2.kolonnas skaitli --> 3
Ievadiet matricas 2.rindas 3.kolonnas skaitli --> 4
Ievadiet matricas 3.rindas 1.kolonnas skaitli --> 2
Ievadiet matricas 3.rindas 2.kolonnas skaitli --> 2
Ievadiet matricas 3.rindas 3.kolonnas skaitli --> 5
det = -15.0
```