**1MPR09\_Simona\_Bļinova sb24037**

**1.uzdevums**

Programma, kas reālize 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)

A computer screen shot of a computer code

AI-generated content may be incorrect.

Testa piemērs(2)

A screenshot of a computer

AI-generated content may be incorrect.

Testa piemērs(3)

A screenshot of a computer program

AI-generated content may be incorrect.

**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)

A screen shot of a computer

AI-generated content may be incorrect.

Testa piemērs(2)

A number on a blue background

AI-generated content may be incorrect.

Testa piemērs(3)

A blue background with white text

AI-generated content may be incorrect.

**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)

A screen shot of a computer screen

AI-generated content may be incorrect.

Testa piemērs(2)

A screen shot of a computer screen

AI-generated content may be incorrect.

Testa piemērs(3)

A screen shot of a computer program

AI-generated content may be incorrect.