

9. praktiskais darbs. 2. semestris

1. uzdevums

Sastādīt programmu, kas nodrošina kvadrātiskas matricas element ievadi, aprēķina visu "diagonāļu", kas paralēlas galvenai diagonālei summu nodrukā uz ekrāna pašu matricu un "tās diagonāļu" elementu summu.

Izvades formāts:

```
1 2 3 |  
4 5 6 | 3  
7 8 9 | 8  
-----  
7 12 | 15
```

Kods:

```
# Programmas nosaukums: Galveno diagonāļu summa matricai  
# 1. uzdevums (1MPR09_Vladislavs_Babaņins)  
# Uzdevuma formulējums: Sastādīt programmu, kas nodrošina kvadrātiskas matricas elementu  
ievadi, aprēķina visu "diagonāļu", kas paralēlas  
# galvenai diagonālei summu nodrukā uz ekrāna pašu matricu un "tās diagonāļu" elementu  
summu.  
# Izvades formāts:  
# 1 2 3 |  
# 4 5 6 | 3  
# 7 8 9 | 8  
# -----  
# 7 12 | 15  
# Programmas autors: Vladislavs Babaņins  
# Versija 1.0
```

```
import numpy
```

```

def ievade(n, m):

    # Prasa lietotājam ievadīt visas matricas elementus. Matricas elementi var būt tikai vesēlie
    # skaitļi.

    # Ja tiek ievadīts nevesels skaitlis vai simbolu virkne, tad prasa ievadīt tikmēr, kamēr
    # neievadīs pareizi.

    # Atgriež NumPy masīvu ar lietotāja ievadītajiem veseliem elementiem.

    # n - matricas (divdimensiju masīva) rindas skaits

    # m - matricas (divdimensiju masīva) kolonnas skaits

    a = numpy.empty((n, m))

    for i in range(n):

        for j in range(m):

            t = input("Ievadiet matricas elementu a(" + str(i) + "," + str(j) + ") ==> ")

            while is_float(t) == False:

                t = input("Kļūda! Ievadītam skaitlim jābūt veselam!\nIevadiet matricas elementu a(" +
                str(i) + "," + str(j) + ") ==> ")

            t = int(t)

            a[i, j] = t

    # Atgriezt NumPy masīvu ar ievadītiem veseliem elementiem

    return a

```

```

def izvade_matricu_ar_diagonalu_summam(a):

    # Atgriež simbolu virkni, kura reprezentē "glītu diagonāļu summu". Labi atspoguļo tikai
    # skaitļus līdz 999.

    # Piemēram:

    # 3 -1 -2 |
    # 2  3  4 | -2
    # 5  6  7 |  3
    # -----
    #  5  8 | 13

    # a - divdimensijas masīvs (kvadrātiska matrica), kurai sameklējam diagonāļu summu
    # izmantojot funkciju "diagonalu_summa"

```

```
# a.ndim # dimensiju skaits
# a.shape # kortežs ar masīva izmēriem
```

```
n = a.shape[0] # x axis
m = a.shape[1] # y axis
s = ""
c = diagonalu_summa(a)
for i in range(n):
    for j in range(m):
        s = s + "{:4d}".format(int(a[i, j]))
    if i == 0:
        s = s + " | \n"
        len1 = len(s) - 3
        continue
    s = s + " | " + str(int((c[-1 * i]))) + "\n"

s = s + " " + "—" * (len1) + "\n"
```

```
k = 0
for i in range(n):
    if i == 0:
        s = s + "  "
        continue
    s = s + "  " + str(int((c[i])))
    k += 1

s = s + " | " + str(int((c[k + 1])))

return s
```

```

def diagonalu_summa(a):

    # Atgriež divdimensijas masīvu, kur labajā un apaksējā stūrī elementi atspoguļo kvadrātiskas
    # matricas diagonāļu summu

    # Ja nav iespējams aprēķināt, tad atgriež "Kļūda"

    # a - divdimensijas masīvs (kvadrātiska matrica)
    n = a.shape[0]
    m = a.shape[1]

    if m == n:

        c = numpy.zeros(2 * n)

        for k in range(1, 2 * n):

            s = 0

            if k <= n:

                # Apakšsumma

                for j in range(k):

                    s = s + a[n - k + j, j]

            else:

                # augšsumma

                for j in range(k - n, n):

                    s = s + a[n + j - k, j]

            c[k] = s

        return c

    else:

        return "Kļūda"

```

```

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 is_whole(n):
    # Pārbauda vai simbolu virkne ir vesels skaitlis vai nav
    # Ja ir vesels skaitlis, tad True. Ja nav tad False.
    # n - simbolu virkne, kuru pārbauda.
    try:
        n = int(n)
    except:
        return False
    else:
        return True
```

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

```
n = input("Ievadiet kvadrātiskas matricas izmēru ==> ")
while is_natural(n) == False:
    n = input("Kļūda! Ievadītā skaitlim jābūt naturālam!\nIevadiet kvadrātiskas matricas izmēru ==> ")
    n = int(n)

a = ievade(n, n)

print("")
print(izvade_matricu_ar_diagonalu_summam(a))
```

Testa piemēri:

1)

```
Ievadiet kvadrātiskas matricas izmēru ==> 3
Ievadiet matricas elementu a(0,0) ==> 1
Ievadiet matricas elementu a(0,1) ==> 2
Ievadiet matricas elementu a(0,2) ==> 3
Ievadiet matricas elementu a(1,0) ==> 4
Ievadiet matricas elementu a(1,1) ==> 5
Ievadiet matricas elementu a(1,2) ==> 6
Ievadiet matricas elementu a(2,0) ==> 7
Ievadiet matricas elementu a(2,1) ==> 8
Ievadiet matricas elementu a(2,2) ==> 9
```

1	2	3		
4	5	6		3
7	8	9		8
<hr/>				
	7	12		15

2)

```
Ievadiet kvadrātiskas matricas izmēru ==> 2
Ievadiet matricas elementu a(0,0) ==> 1
Ievadiet matricas elementu a(0,1) ==> 2
Ievadiet matricas elementu a(1,0) ==> 3
Ievadiet matricas elementu a(1,1) ==> 4
```

1	2		
3	4		2
<hr/>			
	3		5

3)

```
Ievadiet kvadrātiskas matricas izmēru ==> 1
Ievadiet matricas elementu a(0,0) ==> 3
```

3		
<hr/>		
		3

4)

```
Ievadiet kvadrātiskas matricas izmēru ==> 4
Ievadiet matricas elementu a(0,0) ==> 1
Ievadiet matricas elementu a(0,1) ==> 2
Ievadiet matricas elementu a(0,2) ==> 3
Ievadiet matricas elementu a(0,3) ==> 4
Ievadiet matricas elementu a(1,0) ==> 5
Ievadiet matricas elementu a(1,1) ==> 6
Ievadiet matricas elementu a(1,2) ==> 7
Ievadiet matricas elementu a(1,3) ==> 8
Ievadiet matricas elementu a(2,0) ==> 9
Ievadiet matricas elementu a(2,1) ==> 10
Ievadiet matricas elementu a(2,2) ==> 11
Ievadiet matricas elementu a(2,3) ==> 12
Ievadiet matricas elementu a(3,0) ==> 13
Ievadiet matricas elementu a(3,1) ==> 14
Ievadiet matricas elementu a(3,2) ==> 15
Ievadiet matricas elementu a(3,3) ==> 16
```

1	2	3	4		
5	6	7	8		4
9	10	11	12		11
13	14	15	16		21
<hr/>					
	13	23	30		34

5)

```
Ievadiet kvadrātiskas matricas izmēru ==> 5
Ievadiet matricas elementu a(0,0) ==> 5
Ievadiet matricas elementu a(0,1) ==> 1
Ievadiet matricas elementu a(0,2) ==> 1
Ievadiet matricas elementu a(0,3) ==> 1
Ievadiet matricas elementu a(0,4) ==> 1
Ievadiet matricas elementu a(1,0) ==> 1
Ievadiet matricas elementu a(1,1) ==> 1
Ievadiet matricas elementu a(1,2) ==> 1
Ievadiet matricas elementu a(1,3) ==> 1
Ievadiet matricas elementu a(1,4) ==> 1
Ievadiet matricas elementu a(2,0) ==> 1
Ievadiet matricas elementu a(2,1) ==> 1
Ievadiet matricas elementu a(2,2) ==> 1
Ievadiet matricas elementu a(2,3) ==> 1
Ievadiet matricas elementu a(2,4) ==> 1
Ievadiet matricas elementu a(3,0) ==> 1
Ievadiet matricas elementu a(3,1) ==> 1
Ievadiet matricas elementu a(3,2) ==> 1
Ievadiet matricas elementu a(3,3) ==> 1
Ievadiet matricas elementu a(3,4) ==> 1
Ievadiet matricas elementu a(4,0) ==> 1
Ievadiet matricas elementu a(4,1) ==> 1
Ievadiet matricas elementu a(4,2) ==> 1
Ievadiet matricas elementu a(4,3) ==> 1
Ievadiet matricas elementu a(4,4) ==> 1
```

5	1	1	1	1		
1	1	1	1	1		1
1	1	1	1	1		2
1	1	1	1	1		3
1	1	1	1	1		4
<hr/>						
1	2	3	4		9	

6)

```
Ievadiet kvadrātiskas matricas izmēru ==> 3
Ievadiet matricas elementu a(0,0) ==> 3
Ievadiet matricas elementu a(0,1) ==> -1
Ievadiet matricas elementu a(0,2) ==> -1
Ievadiet matricas elementu a(1,0) ==> -2
Ievadiet matricas elementu a(1,1) ==> -3
Ievadiet matricas elementu a(1,2) ==> 3
Ievadiet matricas elementu a(2,0) ==> 4
Ievadiet matricas elementu a(2,1) ==> 5
Ievadiet matricas elementu a(2,2) ==> 6
```

$$\begin{array}{ccc|c} 3 & -1 & -1 & | \\ -2 & -3 & 3 & | \quad -1 \\ 4 & 5 & 6 & | \quad 2 \\ \hline & 4 & 3 & | \quad 6 \end{array}$$

7)

```
Ievadiet kvadrātiskas matricas izmēru ==> trīs
Kļūda! Ievadītam skaitlim jābūt naturālam!
Ievadiet kvadrātiskas matricas izmēru ==> 0
Kļūda! Ievadītam skaitlim jābūt naturālam!
Ievadiet kvadrātiskas matricas izmēru ==> -1
Kļūda! Ievadītam skaitlim jābūt naturālam!
Ievadiet kvadrātiskas matricas izmēru ==> 12.2
Kļūda! Ievadītam skaitlim jābūt naturālam!
Ievadiet kvadrātiskas matricas izmēru ==> 3.5
Kļūda! Ievadītam skaitlim jābūt naturālam!
Ievadiet kvadrātiskas matricas izmēru ==> 3
Ievadiet matricas elementu a(0,0) ==> 12.4
Kļūda! Ievadītam skaitlim jābūt veselam!
Ievadiet matricas elementu a(0,0) ==> trīs
Kļūda! Ievadītam skaitlim jābūt veselam!
Ievadiet matricas elementu a(0,0) ==> -3.2
Kļūda! Ievadītam skaitlim jābūt veselam!
Ievadiet matricas elementu a(0,0) ==> -3
Ievadiet matricas elementu a(0,1) ==> 3
Ievadiet matricas elementu a(0,2) ==> 0
Ievadiet matricas elementu a(1,0) ==> 0
Ievadiet matricas elementu a(1,1) ==> 0
Ievadiet matricas elementu a(1,2) ==> 0
Ievadiet matricas elementu a(2,0) ==> 0
Ievadiet matricas elementu a(2,1) ==> 0
Ievadiet matricas elementu a(2,2) ==> 0
```

$$\begin{array}{ccc|c} -3 & 3 & 0 & \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ \hline & 0 & 0 & -3 \end{array}$$

8)

```
Ievadiet kvadrātiskas matricas izmēru ==> 6
Ievadiet matricas elementu a(0,0) ==> 1
Ievadiet matricas elementu a(0,1) ==> 1
Ievadiet matricas elementu a(0,2) ==> 1
Ievadiet matricas elementu a(0,3) ==> 1
Ievadiet matricas elementu a(0,4) ==> 1
Ievadiet matricas elementu a(0,5) ==> 1
Ievadiet matricas elementu a(1,0) ==> 1
Ievadiet matricas elementu a(1,1) ==> 1
Ievadiet matricas elementu a(1,2) ==> 1
Ievadiet matricas elementu a(1,3) ==> 1
Ievadiet matricas elementu a(1,4) ==> 1
Ievadiet matricas elementu a(1,5) ==> 1
Ievadiet matricas elementu a(2,0) ==> 1
Ievadiet matricas elementu a(2,1) ==> 1
Ievadiet matricas elementu a(2,2) ==> 1
Ievadiet matricas elementu a(2,3) ==> 1
Ievadiet matricas elementu a(2,4) ==> 1
Ievadiet matricas elementu a(2,5) ==> 1
Ievadiet matricas elementu a(3,0) ==> 100
Ievadiet matricas elementu a(3,1) ==> 1
Ievadiet matricas elementu a(3,2) ==> 1
Ievadiet matricas elementu a(3,3) ==> 1
Ievadiet matricas elementu a(3,4) ==> 1
Ievadiet matricas elementu a(3,5) ==> 1
Ievadiet matricas elementu a(4,0) ==> 1
Ievadiet matricas elementu a(4,1) ==> 1
Ievadiet matricas elementu a(4,2) ==> 1
Ievadiet matricas elementu a(4,3) ==> 1
Ievadiet matricas elementu a(4,4) ==> 1
Ievadiet matricas elementu a(4,5) ==> 1
Ievadiet matricas elementu a(5,0) ==> 1
Ievadiet matricas elementu a(5,1) ==> 1
Ievadiet matricas elementu a(5,2) ==> 1
Ievadiet matricas elementu a(5,3) ==> 1
Ievadiet matricas elementu a(5,4) ==> 1
Ievadiet matricas elementu a(5,5) ==> 1
```

1	1	1	1	1	1		
1	1	1	1	1	1		1
1	1	1	1	1	1		2
100	1	1	1	1	1		3
1	1	1	1	1	1		4
1	1	1	1	1	1		5
<hr/>							
1	2	102	4	5			6

9)

```
Ievadiet kvadrātiskas matricas izmēru ==> 3
Ievadiet matricas elementu a(0,0) ==> 3
Ievadiet matricas elementu a(0,1) ==> 100
Ievadiet matricas elementu a(0,2) ==> 3
Ievadiet matricas elementu a(1,0) ==> 4
Ievadiet matricas elementu a(1,1) ==> 5
Ievadiet matricas elementu a(1,2) ==> 6
Ievadiet matricas elementu a(2,0) ==> 7
Ievadiet matricas elementu a(2,1) ==> 8
Ievadiet matricas elementu a(2,2) ==> 9
```

3	100	3		
4	5	6		3
7	8	9		106
<hr/>				
	7	12		17

10)

```
Ievadiet kvadrātiskas matricas izmēru ==> 3
Ievadiet matricas elementu a(0,0) ==> 999
Ievadiet matricas elementu a(0,1) ==> 999
Ievadiet matricas elementu a(0,2) ==> 999
Ievadiet matricas elementu a(1,0) ==> 999
Ievadiet matricas elementu a(1,1) ==> 999
Ievadiet matricas elementu a(1,2) ==> 999
Ievadiet matricas elementu a(2,0) ==> 999
Ievadiet matricas elementu a(2,1) ==> 999
Ievadiet matricas elementu a(2,2) ==> 999
```

999	999	999		
999	999	999		999
999	999	999		1998
<hr/>				
	999	1998		2997

2. uzdevums

Sastādīt programmu, kas realizē teātra biļešu iegādi līdzīgi kā "Biļešu paradīze". Ar šādiem papildus atvieglojumiem:

1. teātra skatītāju zālē ir N rindu un katrā rindā M sēdvietu, M un N vērtības ievāda lietotājs
2. uzsākot iepirkšanos apmēram 50% biļešu ir pārdotas
3. vienā reizē var nopirkt tikai vienu biļeti, ievadot izvēlēto rindas un sēdvietas numuru
4. ja izvēlēta sēdvietā ir brīva, tā tiek atzīmēta kā aizņemta un pāriet pie nākamās biļetes iegādes, bet, ja izvēlēta vieta ir aizņemta, tad uz ekrāna tiek parādīta informācija par visām sēdvietām un pieprasīta atkārtota sēdvietas izvēle
5. Biļešu iegāde tiek atkārtota, kamēr lietotājs ievada 0.rindu un 0.sēdvietu vai visas biļetes ir izpārdotas.

Kods:

```
# Programmas nosaukums: "Biļešu paradīze" (tekstuāls režīms)
```

```
# 2. uzdevums (1MPR09_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas realizē teātra bilešu iegādi līdzīgi kā "Biļešu paradīze".
```

```
# Ar šādiem papildus atvieglojumiem.
```

```
# 1. teātra skatītāju zālē ir N rindu un M sēdvietu. M un N vērtības ievada lietotājs.
```

```
# 2. uzsākot iepirkšanos apmēram 50% biļešu ir pārdotas.
```

```
# 3. vienā reizē var nopirkt tikai vienu biļeti, ievadot izvēlēto rindas un sēdvietas numuru.
```

```
# 4. ja izvēlēta sēdvietā ir brīva, tā tiek atzīmēta kā aizņemta un pāriet pie nākamās biļetes iegādes, bet, ja izvēlēta vieta ir aizņemta,
```

```
# tad uz ekrāna tiek parādīta informācija par visām sēdvietām un pieprasīta atkārtota sēdvietas izvēle.
```

```
# 5. Biļešu iegāde tiek atkārtta, kāmer lietotājs ievada 0.rindu un 0.kolonnun vai visas biļetes ir izpārdotas.
```

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

```
# Versija 1.0
```

```
import numpy
```

```
import random
```

```

def create_random_2array(n, m):

    # Atgriež divdimensiju masīvu, kur visi elementi ir vai nu 0, vai 1 un aptuveni 50% no visu
    # elementu skaita ir 1.

    # n - divdimensiju masīva rindu skaits
    # m - divdimensiju masīva kolonnu skaits (sēdvietu skaits)

    a = numpy.ones((n, m))

    num_ones = int(n * m * 0.5)

    ones_count = 0

    for i in range(n):

        for j in range(m):

            if ones_count < num_ones and random.random() < 0.5:

                a[i][j] = 0

                ones_count += 1

    return a

```

```

def add_numeration_to_matrix(a):

    # Pievieno rindas un kolonnu numerāciju matricai un atgriež simbolu virkni ar matricas
    # elementiem un ir rindas un kolonnas numerācija

    # a - divdimensijas masīvs

    s = ""

    len1 = a.shape[1]

    for j in range(len1 + 1):

        s = s + "{:3d}".format(j) # izmantot formāta norādītāju, lai nodrošinātu nepieciešāmu
        # atstarpus

    s += "\n" # pievienot jaunu rindiņu pēc pirmās rindas

    n = a.shape[0]

    for i in range(n):

        s = s + "{:3d}".format(i + 1) # rindas numurs

        for j in range(len1):

```

```

s = s + " " + a[i, j]

#s = s + "{:3.0f}".format(a[i, j])

s += "\n" # pievienot jaunu rindiņu pēc pirmās rindas

return s

```

```

def choose_what_change_to_one(a, sk):

    # Biļetes pirkšanai. Ja viss ir izpārdots (ja visi ir 1), tad print("Izpārdots!\n")

    # Var izvēlēties, kuru ciparu izmainīt uz 1.

    # 1 -> 1 (vieta aizņemta)

    # 0 -> 1 (print("\nBiļete nopirkta!\n"))

    # a - divdimensijas masīvs (matrica)

    # sk - vieninieku skaits. Ja ir lielāks nekā n x m, tad izpārdots.

    n, m = a.shape

    if sk >= n * m:

        print("\nŠodien viss ir izpārdots!\n")

        return

    while True:

        r = input("Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> ")

        while not is_natural_or_zero(r):

            r = input("Kļūda! Ievadiet vēlamo naturālo rindas numuru biļetes nopirkšanai ==> ")

        r = int(r)

        s = input("Ievadiet vēlamo sēdvietu numuru ==> ")

        while not is_natural_or_zero(s):

            s = input("Kļūda! Ievadiet vēlamo naturālo sēdvietu numuru biļetes nopirkšanai ==> ")

        s = int(s)

```

try:

if r == 0 and s == 0:

return print("Paldies, ka izmantojāt mūsu pakalpojumus!")

elif r == 0 or s == 0:

print("\nNekorekta ievade!\n")

print(add_numeration_to_matrix(replace_ones_to_a_zeros_to_b(a)))

elif a[r - 1, s - 1] == 1:

print("\nJūsu izvēlēta vieta " + str(r) + ".rindā " + str(s) + ".sēdvietā ir aizņemta!\n")

print(add_numeration_to_matrix(replace_ones_to_a_zeros_to_b(a)))

elif a[r - 1, s - 1] == 0:

a[r - 1, s - 1] = 1

print("\nBiļete nopirkta! Jūsu vieta ir " + str(r) + ".rindā " + str(s) + ".sēdvietā.\n")

print(add_numeration_to_matrix(replace_ones_to_a_zeros_to_b(a)))

sk += 1

if sk >= n * m:

print("Šodien viss ir izpārdots!\n")

print(add_numeration_to_matrix(replace_ones_to_a_zeros_to_b(a)))

return

except IndexError:

print("\nIr jāievāda reālu sēdvietu!\nUz redzēšanos!")

quit()

def count_ones(a):

Saskaita vieninieku skaitu noteiktā masīvā a un atgriež to int skaitļi

a - divdimensijas masīvs (matrica)


```
count = 0

for i in range(a.shape[0]):
    for j in range(a.shape[1]):
        if a[i, j] == 1:
            count += 1

return count
```

```
def replace_ones_to_a_zeros_to_b(arr):
    # Izmaina visus divdimensija masīva vieniniekus to A, bet nulles to B
    # 1 -> A
    # 0 -> B
    # arr - divdimensijas masīvs (matrica)
    new_arr = numpy.empty(arr.shape, dtype='str')
    new_arr[arr == 1] = 'A'
    new_arr[arr == 0] = 'B'
    return new_arr
```

```
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 is_natural_or_zero(n):
    # Pārbauda vai simbolu virkne ir naturāls skaitlis vai nulle, vai nav
    # Ja ir naturāls skaitlis vai nulle, 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


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


n = input("Ievadiet kopējo rindu skaitu teātrī ==> ")
while not is_natural(n):
    n = input("Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> ")
n = int(n)


m = input("Ievadiet kopējo sēdvietu skaitu teātrī ==> ")
while not is_natural(m):
    m = input("Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> ")
m = int(m)


a = create_random_2array(n, m)
b = add_numeration_to_matrix(replace_ones_to_a_zeros_to_b(a))


print("")
sk = count_ones(a)
print(b)

print("Sveicinām teātri!\nŠeit var nopirkt biļeti uz izrādi!\n\nIzīesšanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0")

choose_what_change_to_one(a, sk)

```

Testa piemēri:

1)

```
Ievadiet kopējo rindu skaitu teātrī ==> 10
Ievadiet kopējo sēdvietu skaitu teātrī ==> 10
```

0	1	2	3	4	5	6	7	8	9	10
1	B	A	B	B	A	B	B	B	A	B
2	A	B	A	B	B	A	B	A	A	B
3	B	B	B	B	A	A	B	A	A	A
4	B	B	A	B	A	A	B	B	B	A
5	B	B	A	A	B	B	B	B	A	A
6	B	A	B	A	B	A	B	B	A	B
7	B	B	B	B	B	A	A	B	A	A
8	B	B	B	A	B	A	A	A	B	B
9	B	A	A	A	B	A	A	B	A	A
10	A	A	A	A	A	A	A	A	A	A

Sveicinām teātri!

Šeit var nopirkt biļeti uz izrādi!

Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 0

Ievadiet vēlamo sēdvietu numuru ==> 0

Paldies, ka izmantojāt mūsu pakalpojumus!

2)

Ievadiet kopējo rindu skaitu teātrī ==> 10

Ievadiet kopējo sēdvietu skaitu teātrī ==> 9

0	1	2	3	4	5	6	7	8	9
1	A	A	A	B	A	A	A	B	B
2	A	A	A	A	A	B	B	B	A
3	B	A	A	A	B	B	A	A	A
4	A	B	A	A	A	A	B	B	A
5	A	A	A	A	A	A	B	A	A
6	B	A	B	A	A	A	A	B	B
7	B	A	B	A	B	A	A	A	A
8	B	B	B	A	A	B	A	B	A
9	B	B	B	A	B	B	A	A	A
10	A	A	A	A	B	A	B	B	A

Sveicinām teātri!

Šeit var nopirkt biļeti uz izrādi!

Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1

Ievadiet vēlamo sēdvietu numuru ==> 1

Jūsu izvēlēta vieta 1.rindā 1.sēdvietā ir aizņemta!

0	1	2	3	4	5	6	7	8	9
1	A	A	A	B	A	A	A	B	B
2	A	A	A	A	A	B	B	B	A
3	B	A	A	A	B	B	A	A	A
4	A	B	A	A	A	A	B	B	A
5	A	A	A	A	A	A	B	A	A
6	B	A	B	A	A	A	A	B	B
7	B	A	B	A	B	A	A	A	A
8	B	B	B	A	A	B	A	B	A
9	B	B	B	A	B	B	A	A	A
10	A	A	A	A	B	A	B	B	A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 0

Ievadiet vēlamo sēdvietu numuru ==> 0

Paldies, ka izmantojāt mūsu pakalpojumus!

3)

```
Ievadiet kopējo rindu skaitu teātrī ==> 3
Ievadiet kopējo sēdvietu skaitu teātrī ==> 3
```

0	1	2	3
1	A	A	B
2	A	B	A
3	B	A	B

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

```
Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 1
```

Jūsu izvēlēta vieta 1.rindā 1.sēdvietā ir aizņemta!

0	1	2	3
1	A	A	B
2	A	B	A
3	B	A	B

```
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 2
Ievadiet vēlamo sēdvietu numuru ==> 1
```

Jūsu izvēlēta vieta 2.rindā 1.sēdvietā ir aizņemta!

0	1	2	3
1	A	A	B
2	A	B	A
3	B	A	B

```
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 3
```

Biļete nopirkta! Jūsu vieta ir 1.rindā 3.sēdvietā.

0	1	2	3
1	A	A	A
2	A	B	A
3	B	A	B

```
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==>
```

4)

```
Ievadiet kopējo rindu skaitu teātrī ==> 3
Ievadiet kopējo sēdvietu skaitu teātrī ==> 3

  0  1  2  3
  1  B  B  B
  2  A  B  A
  3  A  A  A

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

Izīšanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 1

Biļete nopirkta! Jūsu vieta ir 1.rindā 1.sēdvietā.

  0  1  2  3
  1  A  B  B
  2  A  B  A
  3  A  A  A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 2
Ievadiet vēlamo sēdvietu numuru ==> 2

Biļete nopirkta! Jūsu vieta ir 2.rindā 2.sēdvietā.

  0  1  2  3
  1  A  B  B
  2  A  A  A
  3  A  A  A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 2

Biļete nopirkta! Jūsu vieta ir 1.rindā 2.sēdvietā.

  0  1  2  3
  1  A  A  B
  2  A  A  A
  3  A  A  A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 3

Biļete nopirkta! Jūsu vieta ir 1.rindā 3.sēdvietā.

  0  1  2  3
  1  A  A  A
  2  A  A  A
  3  A  A  A

Šodien viss ir izpārdots!

  0  1  2  3
  1  A  A  A
  2  A  A  A
  3  A  A  A
```

5)

Ievadiet kopējo rindu skaitu teātrī ==> 1
Ievadiet kopējo sēdvietu skaitu teātrī ==> 1

0 1
1 A

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0

Šodien viss ir izpārdots!

6)

```
Ievadiet kopējo rindu skaitu teātri ==> 1
Ievadiet kopējo sēdvietu skaitu teātri ==> 100

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
1 A A B A B B B B B A B B B B B A A A A B A B B A A A B B A A B B B B B A A B A B B B B A A A B A B B B B A A A B A A A A A B B B A A A B A A A A A B B B A A A

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

Izīšanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 1

Jūsu izvēlēta vieta 1.rindā 1.sēdvietā ir aizņemta!

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
1 A A B A A B B B B A B B B B B A A A A B A B B A A A B B A A B B B B B A A B A B B B B A A A B A B B B B A A A B A A A A A B B B A A A B A A A A A B B B A A A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 5

Biļete nopirkta! Jūsu vieta ir 1.rindā 5.sēdvietā.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
1 A A B A A A B B B A B B B B B A A A A B A B B A A A B B A A B B B B B A A B A B B B B A A A B A B B B B A A A B A A A A A B B B A A A B A A A A A B B B A A A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 6

Biļete nopirkta! Jūsu vieta ir 1.rindā 6.sēdvietā.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
1 A A B A A A A B B A B B B B B A A A A B A B B A A A B B A A B B B B B A A B A B B B B A A A B A B B B B A A A B A A A A A B B B A A A B A A A A A B B B A A A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 7

Biļete nopirkta! Jūsu vieta ir 1.rindā 7.sēdvietā.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
1 A A B A A A A B B A B B B B B A A A A B A B B A A A B B A A B B B B B A A B A B B B B A A A B A A A A A B B B A A A B A A A A A B B B A A A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 0

Nekorekta ievade!

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
1 A A B A A A A B B A B B B B B A A A A B A B B A A A B B A A B B B B B A A B A B B B B A A A B A A A A A B B B A A A B A A A A A B B B A A A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 0
Ievadiet vēlamo sēdvietu numuru ==> 0
Paldies, ka izmantojāt mūsu pakalpojumus!
```

7)

```
Ievadiet kopējo rindu skaitu teātrī ==> 2
Ievadiet kopējo sēdvietu skaitu teātrī ==> 2
```

```
0 1 2
1 B B
2 A A
```

```
Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!
```

```
Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 1
```

```
Biļete nopirkta! Jūsu vieta ir 1.rindā 1.sēdvietā.
```

```
0 1 2
1 A B
2 A A
```

```
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 2
Ievadiet vēlamo sēdvietu numuru ==> 2
```

```
Jūsu izvēlēta vieta 2.rindā 2.sēdvietā ir aizņemta!
```

```
0 1 2
1 A B
2 A A
```

```
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 2
```

```
Biļete nopirkta! Jūsu vieta ir 1.rindā 2.sēdvietā.
```

```
0 1 2
1 A A
2 A A
```

```
Šodien viss ir izpārdots!
```

```
0 1 2
1 A A
2 A A
```


8)

Ievadiet kopējo rindu skaitu teātrī ==> 6

Ievadiet kopējo sēdvietu skaitu teātrī ==> 8

0	1	2	3	4	5	6	7	8
1	B	B	B	A	B	B	A	B
2	B	B	A	B	B	A	A	B
3	A	A	A	A	A	A	B	B
4	A	A	A	A	A	B	B	B
5	A	B	B	A	B	A	B	A
6	B	B	A	A	A	A	B	B

Sveicinām teātri!

Šeit var nopirkt biļeti uz izrādi!

Izīšanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 3

Ievadiet vēlamo sēdvietu numuru ==> 2

Jūsu izvēlēta vieta 3.rindā 2.sēdvietā ir aizņemta!

0	1	2	3	4	5	6	7	8
1	B	B	B	A	B	B	A	B
2	B	B	A	B	B	A	A	B
3	A	A	A	A	A	A	B	B
4	A	A	A	A	A	B	B	B
5	A	B	B	A	B	A	B	A
6	B	B	A	A	A	A	B	B

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1

Ievadiet vēlamo sēdvietu numuru ==> 1

Biļete nopirkta! Jūsu vieta ir 1.rindā 1.sēdvietā.

0	1	2	3	4	5	6	7	8
1	A	B	B	A	B	B	A	B
2	B	B	A	B	B	A	A	B
3	A	A	A	A	A	A	B	B
4	A	A	A	A	A	B	B	B
5	A	B	B	A	B	A	B	A
6	B	B	A	A	A	A	B	B

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 0

Ievadiet vēlamo sēdvietu numuru ==> 0

Paldies, ka izmantojāt mūsu pakalpojumus!

9)

```
Ievadiet kopējo rindu skaitu teātrī ==> 3
Ievadiet kopējo sēdvietu skaitu teātrī ==> 3

  0  1  2  3
1  A  A  B
2  B  A  B
3  A  B  A

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> fa
Kļūda! Ievadiet vēlamo naturālo rindas numuru biļetes nopirkšanai ==> 341
Ievadiet vēlamo sēdvietu numuru ==> 3
Ir jāievāda reālu sēdvietu!
Uz redzēšanos!
```

10)

```
Ievadiet kopējo rindu skaitu teātrī ==> 10
Ievadiet kopējo sēdvietu skaitu teātrī ==> 10

  0  1  2  3  4  5  6  7  8  9 10
1  A  A  B  B  B  B  B  A  B  B
2  B  B  A  B  A  A  B  A  B  B
3  A  A  B  B  A  A  B  A  B  B
4  B  A  A  A  B  B  A  B  B  A
5  A  A  B  A  B  B  B  A  A  A
6  B  A  B  B  A  B  A  A  A  B
7  B  A  B  A  B  B  B  B  B  B
8  A  A  A  A  A  A  A  A  A  B
9  B  B  B  B  B  A  B  A  A  B
10 A  B  A  A  A  A  B  A  A  A

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 1

Jūsu izvēlēta vieta 1.rindā 1.sēdvietā ir aizņemta!

  0  1  2  3  4  5  6  7  8  9 10
1  A  A  B  B  B  B  B  A  B  B
2  B  B  A  B  A  A  B  A  B  B
3  A  A  B  B  A  A  B  A  B  B
4  B  A  A  A  B  B  A  B  B  A
5  A  A  B  A  B  B  B  A  A  A
6  B  A  B  B  A  B  A  A  A  B
7  B  A  B  A  B  B  B  B  B  B
8  A  A  A  A  A  A  A  A  A  B
9  B  B  B  B  B  A  B  A  A  B
10 A  B  A  A  A  A  B  A  A  A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 0
Ievadiet vēlamo sēdvietu numuru ==> 0
Paldies, ka izmantojāt mūsu pakalpojumus!
```

11)

```
Ievadiet kopējo rindu skaitu teātrī ==> trīs
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 12.5
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> -3
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 0
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 3.3
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 0.5
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> pieci
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 5
Ievadiet kopējo sēdvietu skaitu teātrī ==> trīs
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 12.5
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> -3
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 0
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 3.3
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 0.5
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> pieci
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 5.0
Kļūda! Ievadiet kopējo naturālu rindu skaitu teātrī ==> 5
```

0	1	2	3	4	5
1	B	B	B	B	A
2	A	A	A	A	B
3	B	A	A	B	A
4	B	B	A	B	B
5	B	A	A	A	A

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

```
Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> -3
Kļūda! Ievadiet vēlamo naturālo rindas numuru biļetes nopirkšanai ==> pieci
Kļūda! Ievadiet vēlamo naturālo rindas numuru biļetes nopirkšanai ==> 0
Ievadiet vēlamo sēdvietu numuru ==> 4
```

Nekorekta ievade!

0	1	2	3	4	5
1	B	B	B	B	A
2	A	A	A	A	B
3	B	A	A	B	A
4	B	B	A	B	B
5	B	A	A	A	A

```
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 7
Ievadiet vēlamo sēdvietu numuru ==> 6
```

Ir jāievāda reālu sēdvietu!
Uz redzēšanos!

12)

```
Ievadiet kopējo rindu skaitu teātrī ==> 3
Ievadiet kopējo sēdvietu skaitu teātrī ==> 3
```

0	1	2	3
1	B	B	B
2	B	A	A
3	A	A	A

```
Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!
```

```
Izīešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> fa
Kļūda! Ievadiet vēlamo naturālo rindas numuru biļetes nopirkšanai ==> 0
Ievadiet vēlamo sēdvietu numuru ==> 13
```

```
Nekorekta ievade!
```

0	1	2	3
1	B	B	B
2	B	A	A
3	A	A	A

```
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 3
Ievadiet vēlamo sēdvietu numuru ==> 5
```

```
Ir jāievāda reālu sēdvietu!
Uz redzēšanos!
```

13)

```
Ievadiet kopējo rindu skaitu teātrī ==> 10
Ievadiet kopējo sēdvietu skaitu teātrī ==> 1

0 1
1 A
2 B
3 B
4 B
5 B
6 B
7 A
8 A
9 A
10 A

Sveicinām teātri!
Šeit var nopirkt biļeti uz izrādi!

Iziešanai no sistēmas ievadiet rindas numuru: 0, sēdvietu: 0
Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 1
Ievadiet vēlamo sēdvietu numuru ==> 1

Jūsu izvēlēta vieta 1.rindā 1.sēdvietā ir aizņemta!

0 1
1 A
2 B
3 B
4 B
5 B
6 B
7 A
8 A
9 A
10 A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 2
Ievadiet vēlamo sēdvietu numuru ==> 1

Biļete nopirkta! Jūsu vieta ir 2.rindā 1.sēdvietā.

0 1
1 A
2 A
3 B
4 B
5 B
6 B
7 A
8 A
9 A
10 A

Ievadiet vēlamo rindas numuru biļetes nopirkšanai ==> 0
Ievadiet vēlamo sēdvietu numuru ==> 0
Paldies, ka izmantojāt mūsu pakalpojumus!
```

3. uzdevums

Sastādīt programmu, kas realizē labirinta izveidi un apstaigāšanu atbilstoši lekcijā dotajiem nosacījumiem, papildus paredzot, ka labirintā 5% no rūtiņu kopskaita ir necaurejami šķēršļi.

Kods:

```
# Programmas nosaukums: Labirints ar necaurejami šķēršļiem.
```

```
# 3. uzdevums (1MPR09_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Sastādīt programmu, kas realizē labirinta izveidi un apstaigāšanu atbilstoši lekcijā dotajiem nosacījumiem,
```

```
# papildus paredzot, ka labirinta 5% no rūtiņu kopskaita ir necaurejami šķēršļi.
```

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

```
# Versija 1.0
```

```
import numpy
```

```
def is_labirint_path(i, j, lab, tr):
```

```
    # Atgriež masīvu found_path (atrasts ceļš), kurš sastāv no frāzem "Pa labi", "Uz leju".
```

```
    # Rekursīvi atrod ceļu, lai izietu no labirinta (var iet tikai no mazākas vērtības uz lielāku (vai vienādu)).
```

```
    # i - 0 (parasti)
```

```
    # j - 0 (parasti)
```

```
    # lab - divdimensijas masīvs ar labirinta segmēnta vērtībām.
```

```
    # tr - trase
```

```
    # trase = numpy.empty(path_length, "O")
```

```
    # path_length = n + m - 2
```

```
    n = lab.shape[0] - 1
```

```
    m = lab.shape[1] - 1
```

```
    if (i == n) and (j == m):
```

```

    return True
else:
    found_path = False
    if (i < n) and (lab[i + 1, j] >= lab[i, j]):
        tr[i + j] = "Uz leju"
        found_path = is_labirint_path(i + 1, j, lab, tr)
    if not found_path and (j < m) and (lab[i, j + 1] >= lab[i, j]):
        tr[i + j] = "Pa labi"
        found_path = is_labirint_path(i, j + 1, lab, tr)
    return found_path

```

```
def izvade(a):
```

```
    # Atgriež simbolu virkni, kura reprezentē noteikto matricu, kā simbolu virkni stābiņveidā.
```

```
    # -1 pārkonvertē kā X.
```

```
    # X - nejauši šķēršļi (-1)
```

```
    # a - divdimensijas masīvs (matrica)
```

```
    # a.ndim # dimensiju skaits
```

```
    # a.shape # kortežs ar masīva izmēriem
```

```
    n = a.shape[0] # x axis
```

```
    m = a.shape[1] # y axis
```

```
    s = "\n"
```

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

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

```
            if a[i, j] == -1:
```

```
                s = s + " X"
```

```
            else:
```

```
                s = s + "{:4d}".format(int((a[i, j])))
```

```
        s = s + "\n"
```

```
return s
```

```
def get_column(a):
```

```
    # a - viendimensijas masīvs
```

```
    # [1, 2, 3] konvertēsies
```

```
    # 1
```

```
    # 2
```

```
    # 3
```

```
    # Atgriež str masīva elementus, kā kolonnu
```

```
    col_str = ""
```

```
    for elem in a:
```

```
        col_str = col_str + str(elem) + "\n"
```

```
    return col_str
```

```
def random_negative_fill(a, chance):
```

```
    # chance = 0.05 (5%) no kopēja skaita būs šķēršļi (-1)
```

```
    # chance - skaitlis (float). Jo lielāka šī vērtība, jo lielāka iespēja, ka būs vairāk "segmētu" ar -
```

1.

```
    # a - divdimensijas masīvs (labirints)
```

```
    # Atgriež divdimensijas masīvu (matricu) kurš ir aizpildīts ar -1.
```

```
    # -1 skaits ir chance * masīva_kopējais_elementu_skaits (chance == 0.05 (5%))
```

```
    a_shape = a.shape
```

```
    # Aprēķināt ievietojamo negatīvo vērtību skaitu
```

```
    num_negatives = int(a_shape[0] * a_shape[1] * chance)
```

```
    # Izvēlieties nejaušus rindu un kolonnu "segmētus", lai ievietotu negatīvas vērtības
```


indices = numpy.zeros((num_negatives, 2), dtype=int) # indices ir nulles masīvs, kura izmērs ir (num_negatives, 2), un tajā tiek glabāti izvēlēto "segmētu" rindu un kolonnu koordinātas.

Pēc nejauši izvēlēto koordinātu noteikšanas row_idx un col_idx, tiek pārbaudīts, vai tās nav jau aizpildītas ar -1, un

vai tās nav sākumpunkts (0, 0) vai beigu punkts (a_shape[0] - 1, a_shape[1] - 1), kas atbilst labirinta sākumam un beigām.

Ja nosacījumi tiek izpildīti, koordinātas tiek ievietotas masīvā indices un while cikls tiek pārtraukts.

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

```
    while True: # Tiek izmantots while cikls, lai izvēlētos "segmētu" koordinātas, kas nav jau iepriekš aizpildītas ar -1.
```

```
        row_idx = numpy.random.randint(a_shape[0])
```

```
        col_idx = numpy.random.randint(a_shape[1])
```

```
        if a[row_idx, col_idx] != -1 and not (row_idx == 0 and col_idx == 0) and not (row_idx == a_shape[0] - 1 and col_idx == a_shape[1] - 1):
```

```
            indices[i] = (row_idx, col_idx)
```

```
            break
```

```
# Aizpildīt nejauši izvēlētos "segmētus" ar -1
```

```
row_indices = indices[:, 0]
```

```
col_indices = indices[:, 1]
```

```
for i in range(len(row_indices)):
```

```
    row_idx = row_indices[i]
```

```
    col_idx = col_indices[i]
```

```
    a[row_idx, col_idx] = -1
```

```
return a
```

```
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 is_natural_or_zero(n):

    # Pārbauda vai simbolu virkne ir naturāls skaitlis vai nulle, vai nav

    # Ja ir naturāls skaitlis vai nulle, 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
```

```
# -----

# Galvenā programmas daļa

# -----
```

```
n = input("Ievadi labirinta garumu ==> ")

while not is_natural(n):

    n = input("Kļūda! Ievadiet naturālu labirinta garumu ==> ")

n = int(n)

m = input("Ievadi labirinta platumu ==> ")

while not is_natural(m):

    m = input("Kļūda! Ievadiet naturālu labirinta platumu ==> ")
```

```
m = int(m)
```

```
labirints = numpy.empty((n, m))
```

```
path_length = n + m - 2
```

```
trase = numpy.empty(path_length, "O")
```

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

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

```
        t = input("Labirints(" + str(i) + "," + str(j) + ") ==> ")
```

```
        while not is_natural_or_zero(t):
```

```
            t = input("Kļūda! Ievadiet naturālu (vai 0) labirinta segmenta vērtību!\nLabirints(" +  
str(i) + "," + str(j) + ") ==> ")
```

```
            labirints[i, j] = t
```

```
print("\nJusu labirints:")
```

```
print(izvade(labirints))
```

```
print("Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):")
```

```
a = random_negative_fill(labirints, 0.05)
```

```
print(izvade(a))
```

```
if is_labirint_path(0, 0, a, trase):
```

```
    print("Labirintā var virzoties tikai no mazākas vērtības uz lielāku (vai vienādu)!\nVirzoties var  
tikai uz leju vai pa labi!")
```

```
    print("Labirintu var iziet virzoties tā:")
```

```
    print(get_column(trase))
```

```
else:
```

```
    print("Labirints nav izejams!")
```

Testa piemēri:

1)

```
Ievadi labirinta garumu ==> 3
Ievadi labirinta platumu ==> 3
Labirints(0,0) ==> 1
Labirints(0,1) ==> 2
Labirints(0,2) ==> 3
Labirints(1,0) ==> 4
Labirints(1,1) ==> 5
Labirints(1,2) ==> 6
Labirints(2,0) ==> 7
Labirints(2,1) ==> 8
Labirints(2,2) ==> 9
```

Jusu labirints:

1	2	3
4	5	6
7	8	9

Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

1	2	3
4	5	6
7	8	9

Labirintā var virzoties tikai no mazākas vērtības uz lielāku (vai vienādu)!

Virzoties var tikai uz leju vai pa labi!

Labirintu var iziet virzoties tā:

Uz leju

Uz leju

Pa labi

Pa labi

2)

```
Ievadi labirinta garumu ==> 3
Ievadi labirinta platumu ==> 3
Labirints(0,0) ==> 1
Labirints(0,1) ==> 2
Labirints(0,2) ==> 3
Labirints(1,0) ==> 4
Labirints(1,1) ==> 5
Labirints(1,2) ==> 6
Labirints(2,0) ==> 7
Labirints(2,1) ==> 8
Labirints(2,2) ==> 9
```

Jusu labirints:

1	2	3
4	5	6
7	8	9

Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

1	2	3
4	5	6
7	8	9

Labirintā var virzoties tikai no mazākas vērtības uz lielāku (vai vienādu)!

Virzoties var tikai uz leju vai pa labi!

Labirintu var iziet virzoties tā:

Uz leju

Uz leju

Pa labi

Pa labi

3)

```
Ievadi labirinta garumu ==> 4
Ievadi labirinta platumu ==> 8
Labirints(0,0) ==> 1
Labirints(0,1) ==> 2
Labirints(0,2) ==> 3
Labirints(0,3) ==> 4
Labirints(0,4) ==> 5
Labirints(0,5) ==> 6
Labirints(0,6) ==> 7
Labirints(0,7) ==> 89
Labirints(1,0) ==> 5
Labirints(1,1) ==> 4
Labirints(1,2) ==> 2
Labirints(1,3) ==> 67
Labirints(1,4) ==> 734
Labirints(1,5) ==> 34
Labirints(1,6) ==> 4
Labirints(1,7) ==> 5
Labirints(2,0) ==> 7
Labirints(2,1) ==> 8
Labirints(2,2) ==> 2
Labirints(2,3) ==>
Kļūda! Ievadiet naturālu (vai 0) labirinta segmēta vērtību!
Labirints(2,3) ==> 3
Labirints(2,4) ==> 0
Labirints(2,5) ==> 6
Labirints(2,6) ==> 3
Labirints(2,7) ==> 1
Labirints(3,0) ==> 12.4
Kļūda! Ievadiet naturālu (vai 0) labirinta segmēta vērtību!
Labirints(3,0) ==> 1
Labirints(3,1) ==> 3
Labirints(3,2) ==> 7
Labirints(3,3) ==> 8
Labirints(3,4) ==> 6
Labirints(3,5) ==> 5
Labirints(3,6) ==> 4
Labirints(3,7) ==> 3

Jusu labirints:

  1  2  3  4  5  6  7  89
  5  4  2  67 734 34  4  5
  7  8  2  3  0  6  3  1
  1  3  7  8  6  5  4  3

Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

  1  2  3  4  5  6  7  89
  5  4  2  67 734 34  4  5
  7  8  2  3  0  6  3  1
  1  3  7  X  6  5  4  3

Labirints nav izejams!
```

4)

```
Ievadi labirinta garumu ==> 6
Ievadi labirinta platumu ==> 1
Labirints(0,0) ==> 1
Labirints(1,0) ==> 2
Labirints(2,0) ==> 3
Labirints(3,0) ==> 4
Labirints(4,0) ==> 5
Labirints(5,0) ==> 6
```

Jusu labirints:

```
1
2
3
4
5
6
```

Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

```
1
2
3
4
5
6
```

Labirintā var virzoties tikai no mazākas vērtības uz lielāku (vai vienādu)!

Virzoties var tikai uz leju vai pa labi!

Labirintu var iziet virzoties tā:

Uz leju

Uz leju

Uz leju

Uz leju

Uz leju

5)

```
Ievadi labirinta garumu ==> 7
Ievadi labirinta platumu ==> 3
Labirints(0,0) ==> 1
Labirints(0,1) ==> 2
Labirints(0,2) ==> 3
Labirints(1,0) ==> 4
Labirints(1,1) ==> 5
Labirints(1,2) ==> 6
Labirints(2,0) ==> 7
Labirints(2,1) ==> 8
Labirints(2,2) ==> 9
Labirints(3,0) ==> 1
Labirints(3,1) ==> 2
Labirints(3,2) ==> 4
Labirints(4,0) ==> 3
Labirints(4,1) ==> 6
Labirints(4,2) ==> 44
Labirints(5,0) ==> 55
Labirints(5,1) ==> 3
Labirints(5,2) ==> 234
Labirints(6,0) ==> 4
Labirints(6,1) ==> 7
Labirints(6,2) ==> 8
```

Jusu labirints:

```
1  2  3
4  5  6
7  8  9
1  2  4
3  6  44
55  3 234
4  7  8
```

Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

```
1  X  3
4  5  6
7  8  9
1  2  4
3  6  44
55  3 234
4  7  8
```

Labirints nav izejams!

6)

```
Ievadi labirinta garumu ---> 8
Kļūda! Ievadiet naturālu labirinta garumu --> 8
Kļūda! Ievadiet naturālu labirinta garumu --> -2
Kļūda! Ievadiet naturālu labirinta garumu --> 12.4
Kļūda! Ievadiet naturālu labirinta garumu --> 12
Ievadi labirinta platumu ---> 3
Labirints(0,0) ---> 1
Labirints(0,1) ---> pieci
Kļūda! Ievadiet naturālu (vai 0) labirinta segmēta vērtību!
Labirints(0,1) ---> 5
Labirints(0,2) ---> -5
Kļūda! Ievadiet naturālu (vai 0) labirinta segmēta vērtību!
Labirints(0,2) ---> 12.4
Kļūda! Ievadiet naturālu (vai 0) labirinta segmēta vērtību!
Labirints(0,2) ---> 12
Labirints(1,0) ---> 12
Labirints(1,1) ---> 1
Labirints(1,2) ---> 1
Labirints(2,0) ---> 1
Labirints(2,1) ---> 0
Labirints(2,2) ---> -1
Kļūda! Ievadiet naturālu (vai 0) labirinta segmēta vērtību!
Labirints(2,2) ---> 2
Labirints(3,0) ---> 3
Labirints(3,1) ---> 5
Labirints(3,2) ---> 6
Labirints(4,0) ---> 3
Labirints(4,1) ---> 3
Labirints(4,2) ---> 2
Labirints(5,0) ---> 1
Labirints(5,1) ---> 6
Labirints(5,2) ---> 7
Labirints(6,0) ---> 5
Labirints(6,1) ---> 3
Labirints(6,2) ---> 88
Labirints(7,0) ---> 6
Labirints(7,1) ---> 4
Labirints(7,2) ---> 2
Labirints(8,0) ---> 88
Labirints(8,1) ---> 546
Labirints(8,2) ---> 4
Labirints(9,0) ---> 5
Labirints(9,1) ---> 6
Labirints(9,2) ---> 7
Labirints(10,0) ---> 8
Labirints(10,1) ---> 2
Labirints(10,2) ---> 2
Labirints(11,0) ---> 2
Labirints(11,1) ---> 2
Labirints(11,2) ---> 1

Jūsu labirints:

  1  5 12
12  1  1
  1  0  2
  3  5  6
  3  3  2
  1  6  7
  5  3 88
  6  4  2
88 546  4
  5  6  7
  8  2  2
  2  2  1

Jūsu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

  1  5 12
12  1  1
  1  0  2
  X  5  6
  3  3  2
  1  6  7
  5  3 88
  6  4  2
88 546  4
  5  6  7
  8  2  2
  2  2  1

Labirints nav izejams!
```

7)

```
Ievadi labirinta garumu ==> 7
Ievadi labirinta platumu ==> 7
Labirints(0,0) ==> 1
Labirints(0,1) ==> 1
Labirints(0,2) ==> 1
Labirints(0,3) ==> 11
Labirints(0,4) ==> 1
Labirints(0,5) ==> 1
Labirints(0,6) ==> 1
Labirints(1,0) ==> 1
Labirints(1,1) ==> 1
Labirints(1,2) ==> 1
Labirints(1,3) ==> 1
Labirints(1,4) ==> 1
Labirints(1,5) ==> 11
Labirints(1,6) ==> 1
Labirints(2,0) ==> 1
Labirints(2,1) ==> 1
Labirints(2,2) ==> 1
Labirints(2,3) ==> 11
Labirints(2,4) ==> 1
Labirints(2,5) ==> 1
Labirints(2,6) ==> 11
Labirints(3,0) ==> 1
Labirints(3,1) ==> 1
Labirints(3,2) ==> 1
Labirints(3,3) ==> 1
Labirints(3,4) ==> 1
Labirints(3,5) ==> 11
Labirints(3,6) ==> 1
Labirints(4,0) ==> 1
Labirints(4,1) ==> 1
Labirints(4,2) ==> 1
Labirints(4,3) ==> 1
Labirints(4,4) ==> 1
Labirints(4,5) ==> 11
Labirints(4,6) ==> 1
Labirints(5,0) ==> 1
Labirints(5,1) ==> 1
Labirints(5,2) ==> 1
Labirints(5,3) ==> 11
Labirints(5,4) ==> 1
Labirints(5,5) ==> 1
Labirints(5,6) ==> 1
Labirints(6,0) ==> 1
Labirints(6,1) ==> 1
Labirints(6,2) ==> 1
Labirints(6,3) ==> 11
Labirints(6,4) ==> 1
Labirints(6,5) ==> 1
Labirints(6,6) ==> 1
```

Jusu labirints:

```
1 1 1 11 1 1 1
1 1 1 1 1 11 1
1 1 1 11 1 1 11
1 1 1 1 1 11 1
1 1 1 1 1 11 1
1 1 1 11 1 1 1
1 1 1 11 1 1 1
```

Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

```
1 1 1 11 1 1 1
1 1 1 1 1 11 1
1 1 X 11 1 1 11
1 1 1 1 1 X 1
1 1 1 1 1 11 1
1 1 1 11 1 1 1
1 1 1 11 1 1 1
```

Labirintā var virzoties tikai no mazākas vērtības uz lielāku (vai vienādu)!

Virzoties var tikai uz leju vai pa labi!

Labirintu var iziet virzoties tā:

Uz leju

Uz leju

Uz leju

Uz leju

Pa labi

Pa labi

Pa labi

Pa labi

Uz leju

Uz leju

Pa labi

Pa labi

8)

```
Ievadi labirinta garumu ==> 0
Kļūda! Ievadiet naturālu labirinta garumu ==> -1
Kļūda! Ievadiet naturālu labirinta garumu ==> pieci
Kļūda! Ievadiet naturālu labirinta garumu ==> 12.5
Kļūda! Ievadiet naturālu labirinta garumu ==> 4
Ievadi labirinta platumu ==> 0
Kļūda! Ievadiet naturālu labirinta platumu ==> -1
Kļūda! Ievadiet naturālu labirinta platumu ==> pieci
Kļūda! Ievadiet naturālu labirinta platumu ==> 12.5
Kļūda! Ievadiet naturālu labirinta platumu ==> 4
Labirints(0,0) ==> -1
Kļūda! Ievadiet naturālu (vai 0) labirinta segmenta vērtību!
Labirints(0,0) ==> 0
Labirints(0,1) ==> 12.5
Kļūda! Ievadiet naturālu (vai 0) labirinta segmenta vērtību!
Labirints(0,1) ==> pieci
Kļūda! Ievadiet naturālu (vai 0) labirinta segmenta vērtību!
Labirints(0,1) ==> trīs
Kļūda! Ievadiet naturālu (vai 0) labirinta segmenta vērtību!
Labirints(0,1) ==> 12.5
Kļūda! Ievadiet naturālu (vai 0) labirinta segmenta vērtību!
Labirints(0,1) ==> 1
Labirints(0,2) ==> 2
Labirints(0,3) ==> 3
Labirints(1,0) ==> 4
Labirints(1,1) ==> 5
Labirints(1,2) ==> 6
Labirints(1,3) ==> 7
Labirints(2,0) ==> 8
Labirints(2,1) ==> 9
Labirints(2,2) ==> 10
Labirints(2,3) ==> 11
Labirints(3,0) ==> 12
Labirints(3,1) ==> 13
Labirints(3,2) ==> 14
Labirints(3,3) ==> 15
```

Jūsu labirints:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Jūsu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Labirintā var virzoties tikai no mazākas vērtības uz lielāku (vai vienādu)!

Virzoties var tikai uz leju vai pa labi!

Labirintu var iziet virzoties tā:

Uz leju

Uz leju

Uz leju

Pa labi

Pa labi

Pa labi

9)

```
Ievadi labirinta garumu ==> 7
Ievadi labirinta platumu ==> 7
Labirints(0,0) ==> 3
Labirints(0,1) ==> 2
Labirints(0,2) ==> 12
Labirints(0,3) ==> 9
Labirints(0,4) ==> 5
Labirints(0,5) ==> 15
Labirints(0,6) ==> 16
Labirints(1,0) ==> 5
Labirints(1,1) ==> 6
Labirints(1,2) ==> 7
Labirints(1,3) ==> 8
Labirints(1,4) ==> 5
Labirints(1,5) ==> 21
Labirints(1,6) ==> 42
Labirints(2,0) ==> 4
Labirints(2,1) ==> 7
Labirints(2,2) ==> 9
Labirints(2,3) ==> 7
Labirints(2,4) ==> 37
Labirints(2,5) ==> 1
Labirints(2,6) ==> 5
Labirints(3,0) ==> 6
Labirints(3,1) ==> 8
Labirints(3,2) ==> 9
Labirints(3,3) ==> 10
Labirints(3,4) ==> 21
Labirints(3,5) ==> 33
Labirints(3,6) ==> 13
Labirints(4,0) ==> 3
Labirints(4,1) ==> 5
Labirints(4,2) ==> 7
Labirints(4,3) ==> 11
Labirints(4,4) ==> 11
Labirints(4,5) ==> 11
Labirints(4,6) ==> 14
Labirints(5,0) ==> 8
Labirints(5,1) ==> 4
Labirints(5,2) ==> 22
Labirints(5,3) ==> 8
Labirints(5,4) ==> 12
Labirints(5,5) ==> 14
Labirints(5,6) ==> 15
Labirints(6,0) ==> 7
Labirints(6,1) ==> 9
Labirints(6,2) ==> 12
Labirints(6,3) ==> 10
Labirints(6,4) ==> 5
Labirints(6,5) ==> 15
Labirints(6,6) ==> 16
```

Jusu labirints:

3	2	12	9	5	15	16
5	6	7	8	5	21	42
4	7	9	7	37	1	5
6	8	9	10	21	33	13
3	5	7	11	11	11	14
8	4	22	8	12	14	15
7	9	12	10	5	15	16

Jusu labirints ar nejaušiem šķēršļiem (5% no kopēja skaita):

3	2	12	9	5	15	16
5	6	7	8	X	21	42
4	7	9	7	37	1	5
6	8	9	10	21	33	13
3	5	7	11	X	11	14
8	4	22	8	12	14	15
7	9	12	10	5	15	16

Labirints nav izejams!

PU1. uzdevums

Realizēt 2. uzdevumu par biļešu iegādi tā, ka uz ekrāna ir redzamas visas sēdvietas un ar peles klikšķi (klikšķiem) var izvēlēties vienu (vairākas) biļetes.

Kods:

```
# Programmas nosaukums: "Biļešu paradīze" ar GUI
```

```
# Papilduzdevums 1 (1MPR09_Vladislavs_Babaņins)
```

```
# Uzdevuma formulējums: Realizēt uzdevumu par biļešu iegādi tā, ka uz ekrāna ir redzamas visas sēdvietas un ar peles klikšķi (klikšķiem) var izvēlēties vienu (vairākas) biļetes.
```

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

```
# Versija 1.0
```

```
import tkinter
```

```
import numpy
```

```
import random
```

```
# global mainīgo saraksts:
```

```
# root, a, label2, buy_button
```

```
def result():
```

```
    # Komanda tiek izsaukta pēc "submit_button" nospiešanas ( = ).
```

```
    # Paņemt n un m no lietotāja un izveido matricu, kur aptuvēni 50% no matricas ir 1.
```

```
    # Izveido globālu mainīgu a un tas ir tās matrica.
```

```
    # Izsauc komandu show_array()
```

```
    global a
```

```
    n = input_rindas_skaits.get()
```

```
    m = input_kolonnas_skaits.get()
```

```
    n = int(n)
```

```
    m = int(m)
```

```

arr = numpy.ones((n, m))

num_ones = int(n * m * 0.5)

ones_count = 0

for i in range(n):
    for j in range(m):
        if ones_count < num_ones and random.random() < 0.5:
            arr[i][j] = 0
            ones_count = ones_count + 1
a = arr
show_array(a)

```

```

def toggle_button(poga):
    # Pogas sēdvietas izvēlēšanai
    # 1 -> 1 "Vieta aizņemta!"
    # 0 -> X "Biļete izvēlēta!"
    # X -> 0 "Biļete atcelta!"

    # check_button_text() pārbauda vai uz visām pogām ir uzrakstīts "1". (pilnā pārlase).
    # Ja uz visiem, tad blokēt pogas un uzrakstīt kā visas biļetes ir izpārdotas.

    if poga["text"] == "0":
        poga["text"] = "X"
        label2.config(text="Biļete izvēlēta!")
        check_button_text()

    elif poga["text"] == "1":
        poga["text"] = "1"
        label2.config(text="Vieta aizņemta!")
        check_button_text()

```

```
elif poga["text"] == "X":  
    poga["text"] = "0"  
    label2.config(text="Biļete atcelta!")  
    check_button_text()
```

```
def change_button_text():  
    # Pārlasa visas pogas root'ā un izmainā "X" uz "1" (izvēlētus uz aizņemts)  
    # Izmainām label2 tekstu uz "Nopirkts!"  
    # Tiek izmantots globāls mainīgais root
```

```
global root
```

```
for poga in root.grid_slaves(): # Parlasam visas pogas  
    if type(poga) == tkinter.Button:  
        if poga.cget("text") == "X": # cget dot iespēju pārbaudīt vai teksts uz pogas ir "X"  
            poga.configure(text="1")
```

```
label2.config(text="Nopirkts!")
```

```
check_button_text()
```

```
def check_button_text():  
    # Izmantojot divus karogus, pārbaudām vai uz visiem pogām ir uzrakstīts "1". (Pogas pilnā pārlase)  
    # Ja uz visām ir "1", tad visas pogas bloķēt un uzrakstīt "Visas biļetes ir izpārdotas!"
```

```
all_ones = True
```

```
has_two = False # two == X (jo pirmkārt tika izmantots 0 - brīvs, 1 - aizņemts, 2 - izvēlēts)
```

```
for poga in root.grid_slaves(): # Pilnā pārlase pogiem un label'iem kuri ir grid'ā izmantojot in
root.grid_slaves() (atrādu internētā tādu funkciju)
```

```
    if poga.cget("text") == "1 - aizņemta vieta\n0 - brīva vieta\nX - izvēlētas vietas
    pirkšanai\nIzvēlieties vietas pirkšanai!":
```

```
        continue
```

```
    if poga.cget("text") == "Nopirkt izvēlētas biļetes":
```

```
        continue
```

```
    if poga.cget("text") == "Nopirkts!":
```

```
        continue
```

```
    if poga.cget("text") == "Biļete izvēlēta!":
```

```
        continue
```

```
    if poga.cget("text") == "X":
```

```
        has_two = True
```

```
    if poga.cget("text") != "1":
```

```
        all_ones = False
```

```
if all_ones:
```

```
    label3 = tkinter.Label(root, text="Visas biļetes ir izpārdotas!", font=("Arial", 12), anchor="ne")
```

```
    label3.place(relx=1.0, rely=0.0, anchor="ne")
```

```
    buy_button.config(state="disabled")
```

```
for poga in root.grid_slaves(): # grid_slave() palīdz visas pogas bloķēt
```

```
    poga.config(state="disabled")
```

```
elif not has_two:
```

```
    buy_button.config(state="disabled")
```

```
else:
```

```
    buy_button.config(state="normal")
```

```
def button_callback(poga):
```


Definē callback() funkciju kas izsauc toggle_button(poga). Jā tieši tā neuzrakstīt, tad programma nestrādās

```
def callback():
```

```
    toggle_button(poga)
```

```
return callback
```

```
def show_array(a):
```

```
    # Izveido jaunu logu "root", un izveido tajā pogas (sēdvietas) nepieciešamā skaitā, ar komandu  
    button_callback
```

```
    # Parada jaunu logu "root".
```

```
    # Tiek izveidota poga "Nopirkt izvēlētas biļetes", pēc tas nospiešanas tiek izsaukta komanda  
    change_button_text
```

```
    # Pogas ir deaktivētas (disabled) pēc noklusējuma.
```

```
    # label2 ir izveidots tukšs un globāls, lai pēc tam to mainītu uz "Nopirkts!" vai uz "".
```

```
    # label4 ir izveidots informatīvam nolūkam
```

```
    # tiek izveidoti trīs globāli mainīgie
```

```
    global root, label2, buy_button
```

```
    root = tkinter.Tk()
```

```
    root.title("Teātris")
```

```
    rindas = len(a)
```

```
    kolonnas = len(a[0])
```

```
    pogas = []
```

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

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

```
            poga = tkinter.Button(root, text=str(int(a[i][j])), width=2, height=1)
```

```
            poga.config(command=button_callback(poga))
```

```
            poga.grid(row=i, column=j)
```

```
            pogas.append(poga)
```

```
label2 = tkinter.Label(root, text="")
```

```
label2.grid(row=i + 1, column=j + 1)
```

```
label4 = tkinter.Label(root, text="1 - aizņemta vieta\n0 - brīva vieta\nX - izvēlētas vietas  
pirkšanai\nIzvēlieties vietas pirkšanai!")
```

```
label4.grid(row=i + 3, column=j + 3, sticky="s")
```

```
buy_button = tkinter.Button(root, text="Nopirkt izvēlētas biļetes", width=20, bd=1,  
command=change_button_text, state="disabled")
```

```
buy_button.grid(row=i + 2, column=j + 2)
```

```
check_button_text()
```

```
root.mainloop()
```

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

```
# Galvenā programmas daļa
```

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

```
logs = tkinter.Tk() # Tkinter (lai izmantotu to komandas)
```

```
logs.geometry("170x130") # Loga izmēra definēšana
```

```
logs.title("Biļešu nopirkšana") # Windows "loga" nosaukums
```

```
# Labels teātrim
```

```
label_teātris = tkinter.Label(logs, text="Teātris", font=("Arial", 12))
```

```
label_teātris.place(x=20, y=6)
```

```
label_teātra_izmers = tkinter.Label(logs, text="Teātra izmērs:", font=("Arial", 12))
```

```
label_teātra_izmers.place(x=20, y=30)
```

```
label_x = tkinter.Label(logs, text="x", font=("Arial", 15))
```

```
label_x.place(x=56, y=58)
```

```
# Entry logi
```

```
input_rindas_skaitis = tkinter.Entry(logs, width=3)
```

```
input_rindas_skaitis.place(x=30, y=65)
```

```
input_kolonnas_skaitis = tkinter.Entry(logs, width=3)
```

```
input_kolonnas_skaitis.place(x=75, y=65)
```

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

```
submit_button = tkinter.Button(logs, text="=", width=20, bd=1, command=result) # Izmantojam  
definētas komandas, lai pēc pogas nospiešanas tā komanda tiek izpildīta
```

```
submit_button.place(x=110, y=62, width=25) # Parādam, kur poga tiks attēlota
```

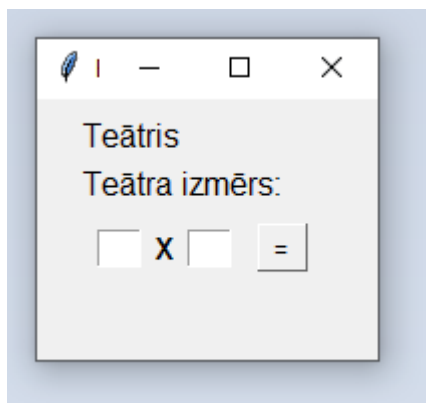
```
logs.mainloop() # lai logs būtu redzāms visu laiku
```

Testa piemēri:

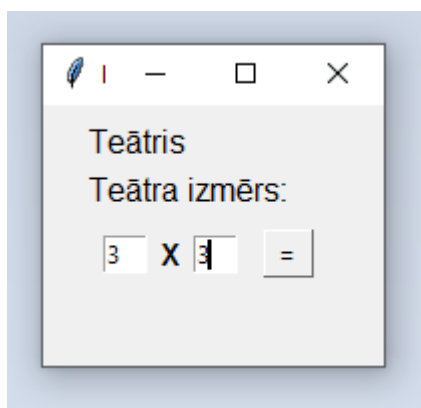
- 1) Video testa piemērs

<https://youtu.be/uvh5oh2g2lQ>

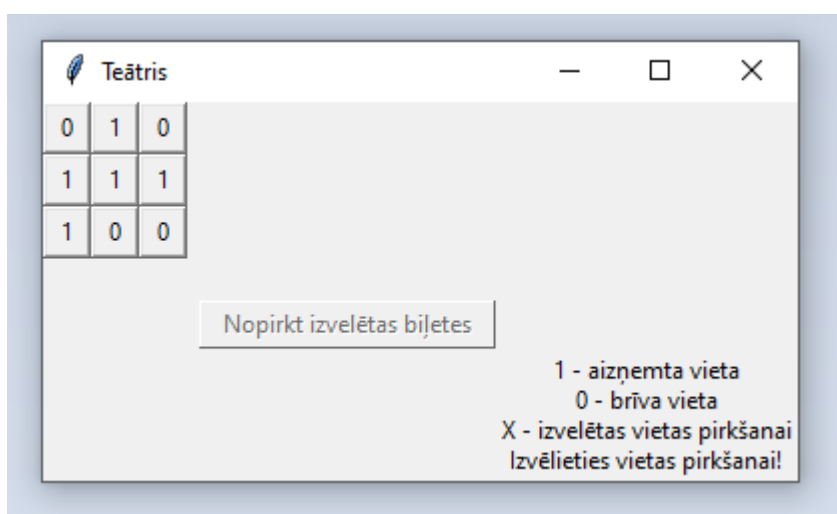
- 2)



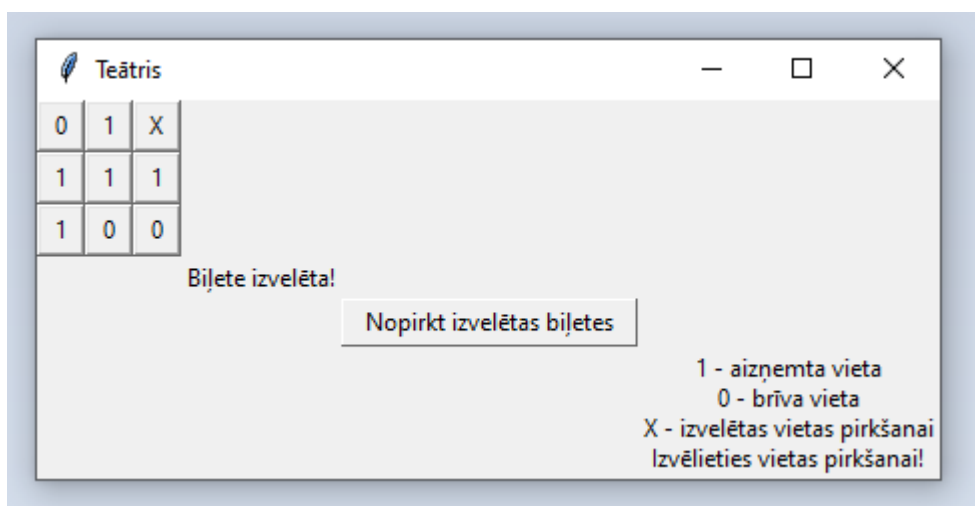
3)



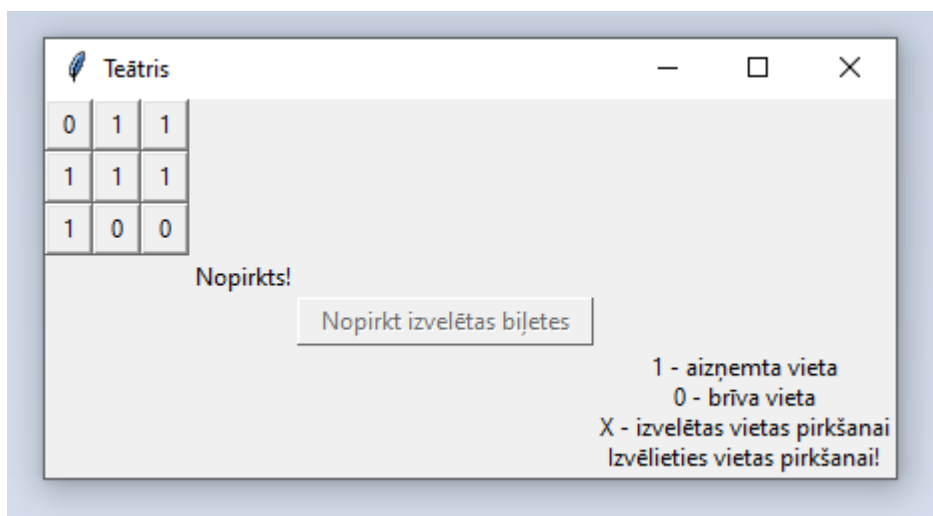
4)



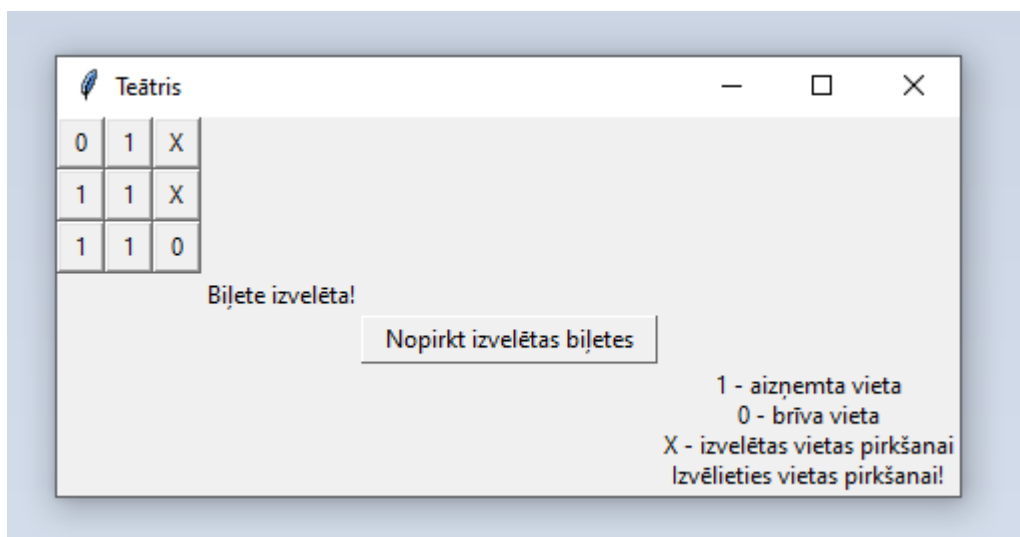
5)



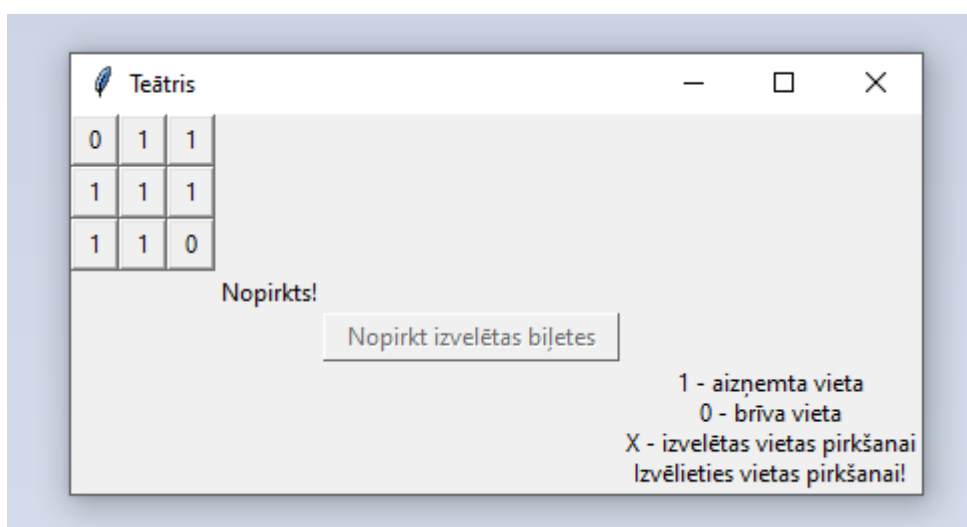
6)



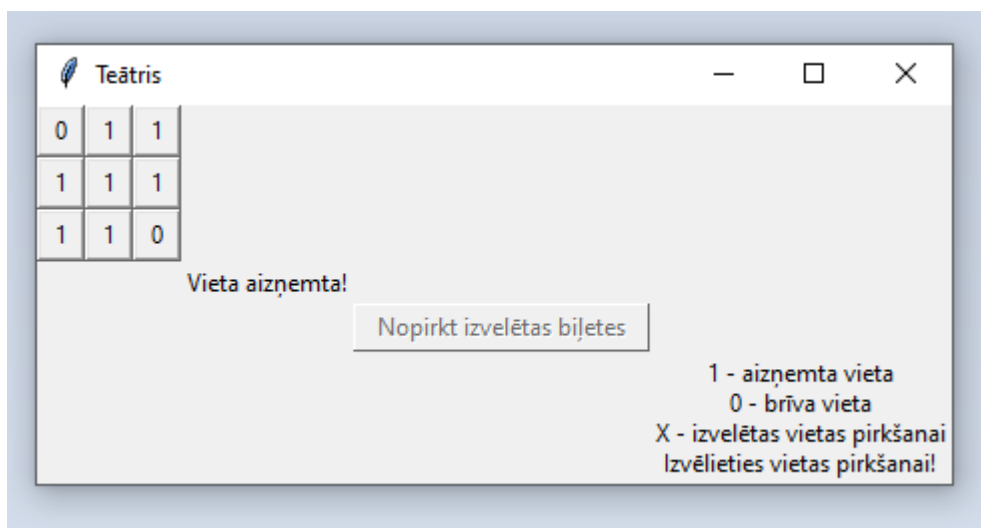
7)



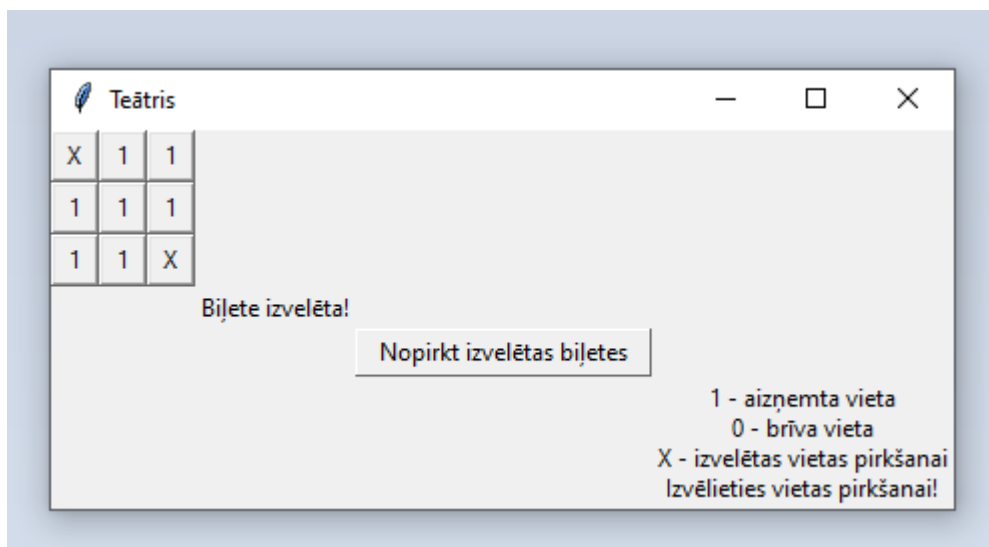
8)



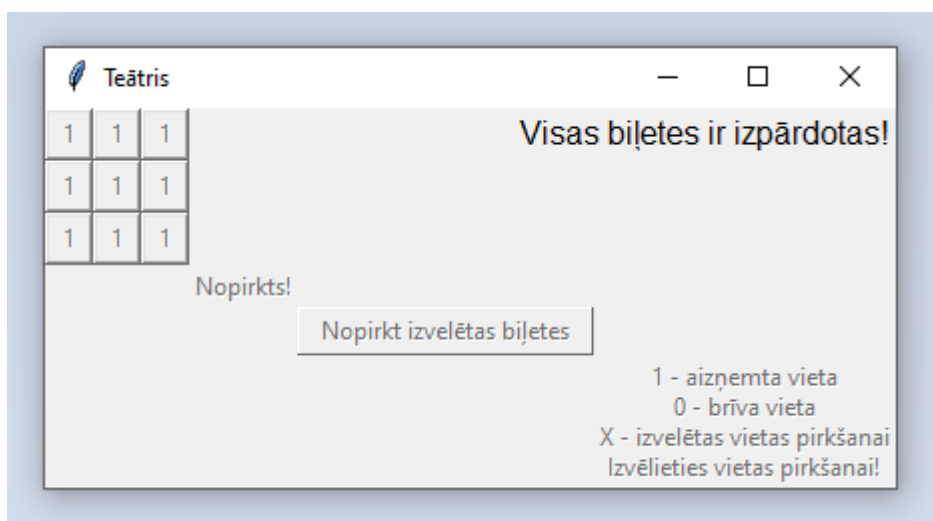
9)



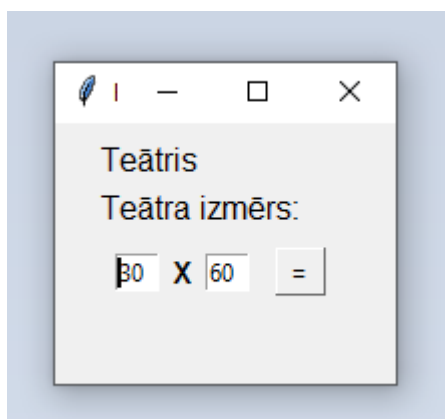
10)



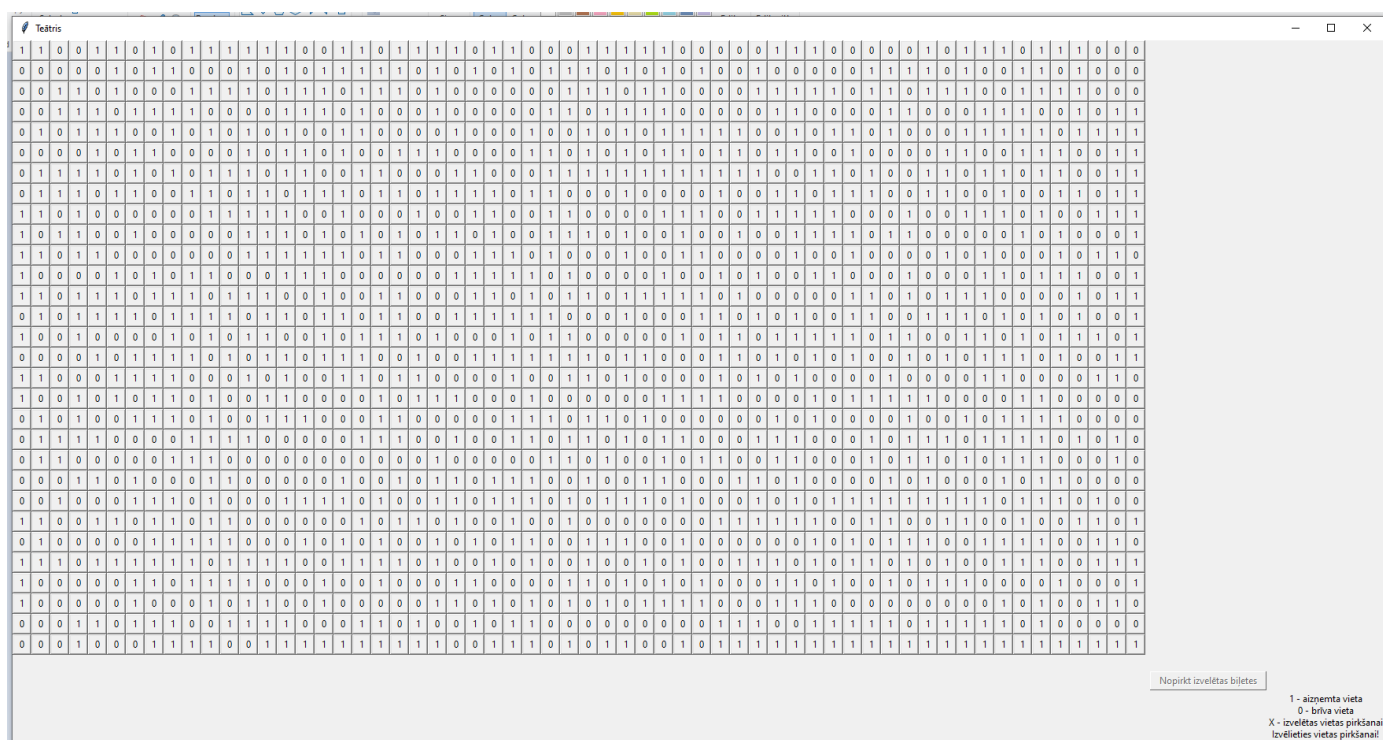
11)



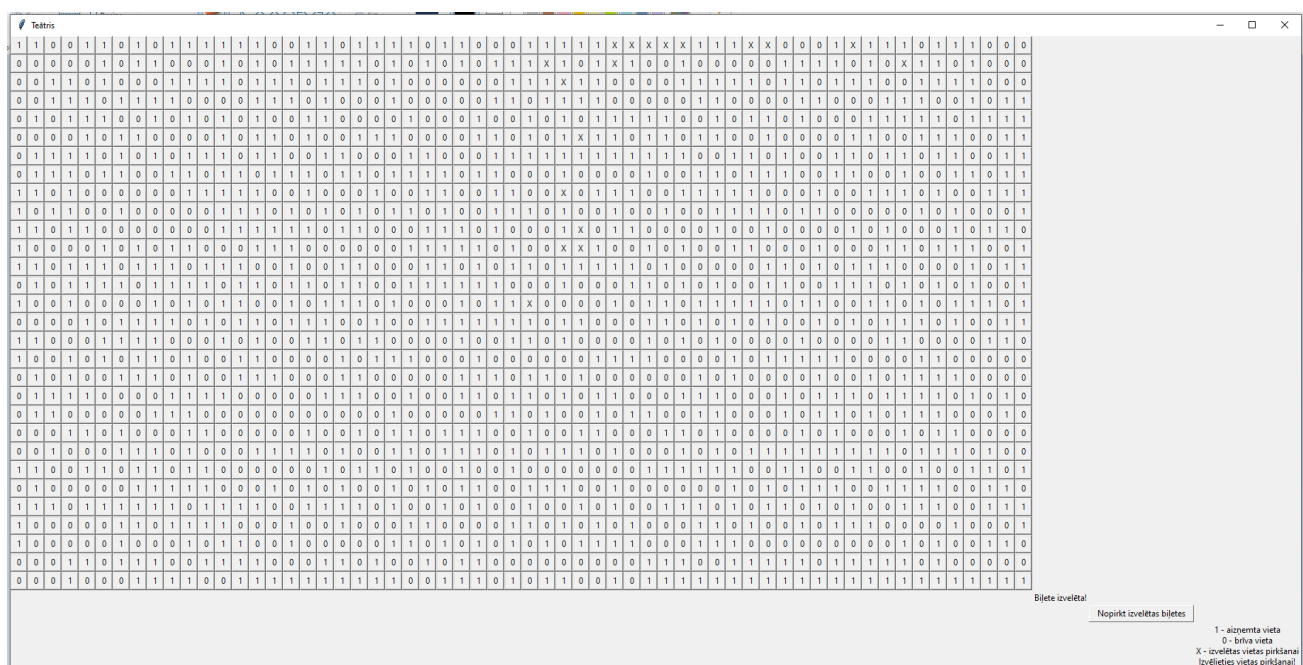
12)



13)



14)



PU2. uzdevums

Realizēt kuģīšu šaušanas spēli jeb spēli KARTUPELIS – 2 spēlētāji dators un lietotājs.

1. Līmenis abu spēlētāju kuģu izvietojumu ievada lietotājs un uz ekrāna redzamas tikai lietotāja gājieni datora “jūrā” un lietotāja “jūra ar kuģiem” un datora izdarītiem gājieniem. Katrs secīgi var izdarīt tikai vienu gājienu.
2. Līmenis – dators pats izvieto savus kuģus “jūrā”
3. Līmenis – dators šāvienus izdara plānveidīgi nevis uz labu laimi.

Kods:

Testa piemēri:

1)