1MPR10_Simona_Bļinova sb24037

1.uzdevums

Programma, kas realizē Latloto 5 no 35 izlozi.

Kods:

```python
import random

def izloze():
    l = set()
    for i in range(5):
        while True:
            x = random.randint(1, 35)
            if not x in l:
                l.add(x)
                break
    return l


def bilete():
    b = set()
    for i in range(5):
        while True:
            x = int(input(f'Ievadiet {i+1} skaitli --> '))
            if not x < 1 and not x > 35:
                if not x in b:
                    b.add(x)
                    break
    return b


b = bilete()
l = izloze()
#print(b, l)

skel = b & l
#print(skel)

match len(skel):
    case 3:
        laimests = 'Jūs atminējat 3 skaitļus, jūs saņēmāt MAZO LAIMESTU!'
```

```
    case 4:
        laimests = 'Jūs atminējat 4 skaitļus, jūs saņēmāt VIDĒJO LAIMESTU!'
    case 5:
        laimests = 'Jūs atminējat 5 skaitļus, jūs saņēmāt LIELO LAIMESTU!'
    case _:
        laimests = 'Jums NAV LAIMESTA!'
```

```
print(' ')
print(f'Izlozētie skaitļi: ', l)
print(f'Jūsu skaitļi: ', b)
print(laimests)
```

Testa piemērs(1)

```
Ievadiet 1 skaitli --> 2
Ievadiet 2 skaitli --> 31
Ievadiet 3 skaitli --> 17
Ievadiet 4 skaitli --> 23
Ievadiet 5 skaitli --> 5

Izlozētie skaitļi:  {34, 11, 12, 21, 22}
Jūsu skaitļi:  {2, 5, 17, 23, 31}
Jums NAV LAIMESTA!
```

Testa piemērs(2)

```
Ievadiet 1 skaitli --> 23
Ievadiet 2 skaitli --> 13
Ievadiet 3 skaitli --> 1
Ievadiet 4 skaitli --> 7
Ievadiet 5 skaitli --> 32

Izlozētie skaitļi:  {32, 3, 4, 11, 12}
Jūsu skaitļi:  {32, 1, 7, 13, 23}
Jums NAV LAIMESTA!
```

Testa piemērs(3)

```
Ievadiet 1 skaitli --> 24
Ievadiet 2 skaitli --> 21
Ievadiet 3 skaitli --> 5
Ievadiet 4 skaitli --> 11
Ievadiet 5 skaitli --> 19

Izlozētie skaitļi:  {6, 11, 18, 25, 27}
Jūsu skaitļi:  {5, 11, 19, 21, 24}
Jums NAV LAIMESTA!
```

2.uzdevums

Programma, kas reālize 4 spēlētāju karšu izvadi.

Kods:

```python
import random

k = set()

for i in range(4):
    kartis = f'{i+1}.spēlētāja kartis: '
    for j in range(6):
        while True:
            # 52 kartis --> 0 - 51
            x = random.randint(0, 51)
            if not x in k:
                k.add(x)
                ks = x // 13
                ksk = x % 13

                match ksk:
                    case 9:
                        kartis += chr(9824+ks)+'J '
                    case 10:
                        kartis += chr(9824+ks)+'Q '
                    case 11:
                        kartis += chr(9824+ks)+'K '
                    case 12:
                        kartis += chr(9824+ks)+'A '
                    case _:
                        kartis += chr(9824+ks)+str(ksk+2)+' '

                break

    print(kartis)
```

Testa piemērs(1)

```
1.spēlētāja kartis: ♣J ♠A ♡8 ♠9 ♡J ♡10
2.spēlētāja kartis: ♠5 ♣4 ♣6 ◇5 ◇4 ◇J
3.spēlētāja kartis: ◇8 ♡3 ♠Q ♠2 ♣9 ♣2
4.spēlētāja kartis: ◇10 ♣4 ♣5 ♡6 ♠8 ♠10
```

Testa piemērs(2)

```
1.spēlētāja kartis: ♠6 ◇8 ♡7 ♠K ♣3 ◇2
2.spēlētāja kartis: ◇10 ♡J ♡3 ♡2 ♠4 ◇7
3.spēlētāja kartis: ♠J ◇5 ♠5 ♣J ♣6 ◇A
4.spēlētāja kartis: ♣3 ♠9 ♡K ♡10 ♣A ◇J
```

Testa piemērs(3)

```
1.spēlētāja kartis: ◇Q ♡5 ◇7 ♣7 ♠6 ♠8
2.spēlētāja kartis: ♠4 ♣2 ♣4 ♡A ♣6 ◇4
3.spēlētāja kartis: ♠9 ◇A ♣3 ♣8 ◇6 ◇8
4.spēlētāja kartis: ♠Q ♣9 ♡8 ♠A ♡4 ◇3
```

3.uzdevums

Programma, kas pārbauda vai lietotāja ievadīts sudoku ir pareizi atrisināts.

Kods:

```python
import numpy
import random

def sudoku():
    c = numpy.empty((9, 9))

    for i in range(9):
        for j in range(9):
            a = int(input(f'Ievadiet skaitli {i+1}.rindā {j+1}.kolonnā --> '))
            c[i, j] = a

    return c


def izvade(a):
    for i in range(9):
        v = ''
        for j in range(9):
            v += str(round(a[i, j])) + ' '
        print(v)


def parbaude(a):
    if rindas(a) == False:
        return False
    if kolonnas(a) == False:
        return False
```

```python
        if kvadrati(a) == False:
            return False
    return True



def rindas(a):
    for i in range(9):
        b = set()
        for j in range(9):
            b.add(a[i, j])
        if len(b) != 9:
            #print('r')
            #print(i)
            #print(len(b))
            return False
    return True



def kolonnas(a):
    for i in range(9):
        b = set()
        for j in range(9):
            b.add(a[j, i])
        if len(b) != 9:
            #print('k')
            #print(j)
            #print(len(b))
            return False
    return True



def kvadrati(a):
    for i in range(0, 9, 3):
        for j in range(0, 9, 3):
            b = set()
            for k in range(0+i, 3+i, 1):
                for p in range(0+j, 3+j, 1):
                    b.add(a[k, p])
            if len(b) != 9:
                #print('kv')
                #print(i, j)
```

```
            #print(len(b))
            return False
    return True


s = sudoku()

print(' ')

izvade(s)

print(' ')

paz = parbaude(s)
#print(paz)

if paz == True:
    print('Sudoku IR atrisināts pareizi.')
else:
    print('Sudoku NAV atrisināts pareizi.')
```

Testa piemērs(1)

```
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9

Sudoku NAV atrisināts pareizi.
```

Testa piemērs(2)

```
1 3 5 6 2 7 4 8 9
9 4 6 8 1 3 7 5 2
7 2 8 5 9 4 3 1 6
2 6 9 4 5 1 8 7 3
5 8 1 7 3 6 9 2 4
4 7 3 9 8 2 1 6 5
8 1 2 3 6 9 5 4 7
6 9 4 1 7 5 2 3 8
3 5 7 2 4 8 6 9 1

Sudoku IR atrisināts pareizi.
```

Testa piemērs(3)

```
9 4 6 8 3 1 2 5 7
7 2 8 5 4 9 6 1 3
1 3 5 6 7 2 9 8 4
8 1 2 3 9 6 7 4 5
6 9 4 1 5 7 8 3 2
3 5 7 2 8 4 1 9 6
2 6 9 4 1 5 3 7 8
4 7 3 9 2 8 5 6 1
5 8 1 7 6 3 4 2 9

Sudoku IR atrisināts pareizi.
```