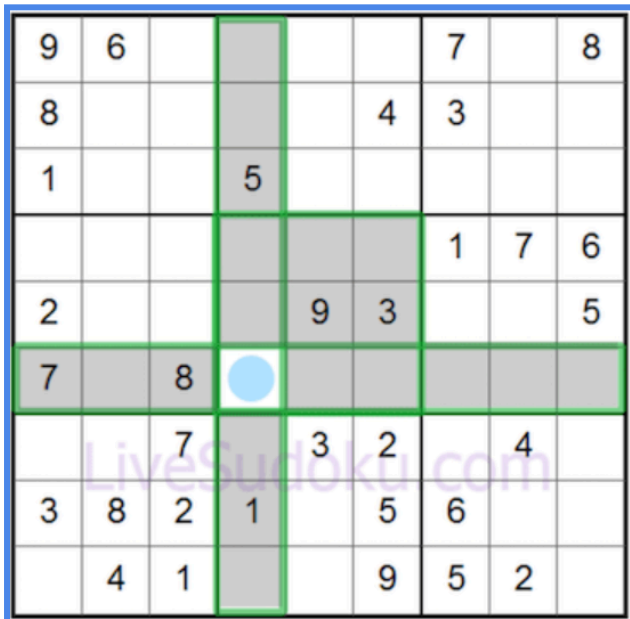


Igra 10



```
*****
**          S U D O K U          **
*****

Pravila:
1.Svaka kolona ima brojeve od 1 do 9.
2.Svaki red ima brojeve od 1 do 9.
3.Svaki 3x3 kvadrat ima brojeve od 1 do 9.
4.Igrac ima pravo da pogresi 3 puta.
5.Brojevi se pogadjaju: 9-2,3 (broj 9 u redu 2 i u koloni 3).
  1 2 3  4 5 6  7 8 9
  =====
1||5 . 1 | . 7 2 | 4 . 9 ||
2||. 4 9 | . 3 . | 7 5 2 ||
3||8 . . | 5 . 9 | . . 6 ||
  -----+-----+-----
4||. 9 . | . 5 7 | . 3 8 ||
5||. . 8 | . 9 . | 5 . 7 ||
6||7 5 3 | . 8 4 | . . . ||
  -----+-----+-----
7||3 . . | 9 2 . | 1 6 5 ||
8||9 6 . | . 1 . | 8 7 3 ||
9||1 8 . | 7 . 3 | . . . ||
```

- Kako igrice logicki radi, redovi, kolone i 3x3 kvadrati
- Napraviti 3 txt fajla sa vec unapred definisanim sudokuima

531984762
649257381
827613459
962378145
185429673
374561298
218745936
753196824
496832517

378415692
429763815
561928734
984672351
613549287
257831469
832157946
745396128
196284573

531672489
649831752
827549316
496157238
218396547
753284691
374928165
962415873
185763924

Implementacija prvog dela:

```
import random

def izaberiFajl():
    r_broj = random.randint(1,3)
    fajl = "example_{}.sudoku".format(r_broj)
    return fajl

def procitajIzfajla():
    f = open(izaberiFajl(), "r")
    s = f.readline()
    lista = []
```

```

while s!="":
    pom = []
    for i in range (9):
        pom.append(s[i])
    lista.append(pom)
    s = f.readline()
f.close()
return lista

```

```

class Sudoku():

```

```

    def __init__(self):
        self.h_lista = procitajIzfajla() # original
        self.igra = [] # igra koju korisnik vidi
        for i in range (9):
            list = []
            self.igra.append(list)
            for j in range (9):
                self.igra[i].append(self.h_lista[i][j])
        for i in range (50):
            random_red = random.randint(0, 8)
            random_kolona = random.randint(0, 8)
            if (self.igra[random_red][random_kolona] != 0):
                del self.igra[random_red][random_kolona]
                self.igra[random_red].insert(random_kolona, '.')

```

```

    def ispisi_sudoku(self, opt=1):

```

```

        if opt == 1:
            # defaultno printaj igru
            lista = self.igra
        else:
            lista = self.h_lista

        print("  1 2 3   4 5 6   7 8 9")
        print("  =====")
        for i in range (9):
            pom = ""
            for j in range (9):
                if j in (2, 5):
                    pom += str(lista[i][j]) + " | "
                else:
                    pom += str(lista[i][j]) + " "
            pom = "{}||".format(i+1) + pom + "||"
            print(pom)
            if (i in (2, 5)):
                print("  -----+-----+-----")
        print("  ===== ")

```

```

    def iscrtaj_naslov(self):

```

```

        print("*****")
        print("**                               **")
        print("**          S U D O K U          **")
        print("**                               **")

```

```
print("*****")

igrica = Sudoku()
igrica.iscrtaj_naslov()
igrica.ispisi_sudoku()
igrica.ispisi_sudoku(2)
broj_promasaja = 0
max_broj_promasaja = 3
```

OBJASNJENJA:

NAPOMENA 1: Razumeti stvarnu igru kako radi

NAPOMENA 2: Sta ce kod nas biti potrebno od funkcija

NAPOMENA 3: Upotreba klasa za implementaciju i onda sva ponasanja sudokua cemo uvaciti tu

NAPOMENA 4: Funkciju za ispis sudoka, objasniti sa opcionim parametrom.

Implementacija drugog dela:

1. DODAJEMO GLAVNI PROGRAM SA PETLJOM:

```
igrica = Sudoku(1)
igrica.iscrtaj_naslov()
igrica.ispisi_sudoku()

broj_promasaja = 0
max_broj_promasaja = 3
kraj_igre = False

while kraj_igre == False:
    if (igrica.krajIgre()):
        print("Bravo! Resio si Sudoku!")
        kraj_igre = True
    if broj_promasaja < max_broj_promasaja :
        (b, r, k) = igrica.unesiBroj(broj_promasaja, max_broj_promasaja)
        ishod = igrica.proveriBroj(b, r, k)
        if ishod == 0:
            broj_promasaja += 1
            print("Nazalost, pogresio si!")
        elif ishod == -2:
            print("Nedozvoljen unos!")
        elif ishod == 1:
            igrica.update(b, r, k)
            print("Super, pogodio si!")
    else:
        print("3 promasaja :( Izgubio siiii...")
        kraj_igre = True
    igrica.ispisi_sudoku()
```

2. DODAVANJE FUNKCIJE ZA UNOS BROJA:

```

def unesiBroj(self, promasaji, ukupno):
    print("Unes je u formatu 1:52 (1-broj koji se unosi, 5-red, 2-kolona)")
    print("Imas {} promasaja od dozvoljenih {}".format(promasaji, ukupno))
    unos = input("Sledeci unos: ")
    (broj, adresa) = unos.split(":")
    red = adresa[0]
    kolona = adresa[1]
    return (int(broj), int(red), int(kolona))

```

3. DODAVANJE FUNKCIJE ZA PROVERU UNOSA U IGRICU:

```

def proveriBroj(self, broj, red, kolona):    #return 0 greska, 1 ako je uspeh
    if red > 9 or kolona > 9:
        return -2    #nedozvoljen unos
    else:
        element = int(self.h_lista[red-1][kolona-1])
        element_igrice = self.igra[red-1][kolona-1]    #ovo treba da je '.'
        if broj == element and element_igrice == '.':
            return 1    # pogodak
        elif broj != element and element_igrice == '.':
            return 0    # promasaj
        elif broj == element and element_igrice != '.':
            return -2 # nedozvoljen unos
        else:
            return -2

```

4. DODAVANJE UPDATE FUNKCIJE

```

def update(self, broj, red, kolona):        # promeni broj u self.igra
    del self.igra[red-1][kolona-1]
    self.igra[red-1].insert(kolona-1, broj)

```

5. DETEKTOVANJE KRAJA IGRE - pobede

```

def krajIgre(self):
    brojac = 0
    for i in range (8):
        for j in range (8):
            if self.igra[i][j] == '.':
                brojac +=1
    if brojac > 0:
        return False
    else:
        return True

```

6. POBOLJSANJE ISPISA SUDOKA (SA OPCIONIM ARGUMENTOM)

```

def ispisSudokua(self, sud = 1):
    lista = self.prazan
    if sud != 1:
        lista = self.popunjen
    print(" 1 2 3   4 5 6   7 8 9")

```

```

print(" =====")
for i in range (9): # petlja za redove!
    pom = ""
    for j in range (9): # unutar jednog reda!
        if j in (2, 5):
            pom += str(lista[i][j]) + " | "
        else:
            pom += str(lista[i][j]) + " "
    pom = "{}||".format(i+1) + pom + "||" # i+1 umjesto
i
    print(pom)
    if i in (2, 5):
        print(" -----+-----+-----")
print(" =====")

```

Adekvatan poziv te funkcije onda:

```

if (igra.krajIgre()):
    igra.ispisSudokua()
    print("Bravo! Resio si Sudoku!")
    kraj_igre = True
if broj_promasaja < max_broj_promasaja :
    (b, r, k) = igra.unesiBroj(broj_promasaja, max_broj_promasaja)
    ishod = igra.proveriBroj(b, r, k)
    if ishod == 0:
        broj_promasaja += 1
        print("Nazalost, pogresio si!")
    elif ishod == -2:
        print("Nedozvoljen unos!")
    elif ishod == 1:
        igra.update(b, r, k)
        print("Super, pogodio si!")
        igra.ispisSudokua()
else:
    print("3 promasaja :( Izgubio siiii...")
    igra.ispisSudokua(0)
    kraj_igre = True

```

7. DODAVANJE LEVELA TEZINE

```

# glavni program
print("Unesi level koji zelis za Sudoku igru:")
level = int(input("1-lak, 2-srednji, 3-tezak"))
igra = Sudoku(level)

```

```

# edit klase
class Sudoku:
    def __init__(self, level):
        self.prazan = []
        self.popunjen = procitajIzFajla()
        self.level = level
        ...
        tezina = generisiTezinu(level)

    for i in range(tezina):
        random_red = random.randint(0, 8)
        random_kolona = random.randint(0, 8)
        ...

def generisiTezinu(level):
    tezine = {1:50, 2:60, 3:70}
    return tezine[level]

```

8. VERIFIKACIJA UNOSA

```

def unesiBroj(self, promasaji, ukupno):
    ...
    while True:
        unos = input("Unos: ")
        if len(unos) != 4:
            print("Neispravan unos.")
        else:
            (broj, adresa) = unos.split(":")
            red = adresa[0]
            kolona = adresa[1]
            if int(broj) > 9 or int(broj) < 1 or int(red) > 9 or
int(red) < 0 or int(kolona) > 9 or int(kolona) < 0:
                print("Neispravan unos.")
            else:
                red = int(red)
                kolona = int(kolona)
                broj = int(broj)
                break
    return (broj, red, kolona)

```