

Formulacija problema

```
77 def proverikraj(igrac,tabla: list[list], x:int, y:int):
78     imaPotez = 0
79     localX = x
80     localY = y
81     if igrac == 0:
82         for i in range(localY):
83             for j in range(localX-1):
84                 if(tabla[i][j] == Nista and tabla[i][j+1] == Nista):
85                     imaPotez+=1
86     else:
87         for i in range(localY-1):
88             for j in range(localX):
89                 if(tabla[i][j] == Nista and tabla[i+1][j] == Nista):
90                     imaPotez+=1
91     if imaPotez == 0:
92         print("Kraj igre za "+igrac+"!")
93         return imaPotez
94     else:
95         print(imaPotez)
96         return imaPotez
```

Funkciji proverikraj prosledjujemo sledece parameter: igrac (igrac za koga treba se radi procena), trenutno stanje tabele I n I m kordinate kako bi prolazio kroz for petlju. Funkcija prolazi kroz svaki moguci potez I dodaje u brojac imaPotez I ako je taj brojac jednak 0 onda znaci da igrac nema vise poteza I samim tim dolazi do kraja igre.

```

14 def odigrajPotez(igr):
15     potez = input("Unesite potez igraca "+igr+": ")
16     xKoordinata = potez[0]
17     yKoordinata = potez[1]
18     xKoordinata2 = abs(ord(xKoordinata)-48-valY)
19     yKoordinata2 = ord(yKoordinata)-65
20     return proveripotez(xKoordinata2,yKoordinata2, igr)

```

OdigrajPotez:

Igrac unosi svoj potez npr. 1D, gde u funkciji xKoordinata dobija vrednost 1, a yKoordinata dobija vrednost D. Posle toga se te kordinate obradjuju i pretvaraju jer brojevi u tabeli idu u opadajucem redosledu, takodje moze se videti da trazimo ascii kod i oduzimamo ga kako bi dobili poziciju na koje treba da se postavi potez.

```

22 def proveripotez(x:int, y:int, igr)->bool:
23     prom = False
24     if(igr==X):
25         if(x>0 and x<valY and y>=0 and y<valX and tabla[int(x-1)][int(y)]==Nista and tabla[int(x)][int(y)]==Nista):
26             tabla[int(x)][int(y)]=igr
27             tabla[int(x-1)][int(y)]=igr
28             prom = True
29             #igrac = 'O'
30     else:
31         if(x>=0 and x<valY and y>=0 and y<valX-1 and tabla[int(x)][int(y)]==Nista and tabla[int(x)][int(y+1)]==Nista):
32             tabla[int(x)][int(y)]=igr
33             tabla[int(x)][int(y+1)]=igr
34             prom = True
35             #igrac = X
36     if prom==False:
37         print("Nevalidan potez!")
38     print_table(tabla, int(valX), int(valY))
39     return prom

```

proveriPotez(x,y, igrac)

Promenljiva "prom" je postavljena na false i proverava da li je uspesno izvršen potez. If petlja sadrži kod koji se izvršava ako je igrač X na potezu, a else kod za igrača O. U if petlji se nalazi još jedna if petlja koja proverava redom da li je x koordinata u dozvoljenom opsegu uzimajući u obzir da i polje iznad mora da bude prazno. Dalje proverava za oba polja gde je potrebno da figura bude postavljena (to su izabrano polje i polje iznad za igrača X, a izabrano polje i polje desno od njega za igrača O). Ukoliko je uslov ispunjen u ta se polja upisuje oznaka igrača (X ili O) i "prom" se postavlja na

True jer je potez validan. fja "print_table" iscrtava novo stanje tabele. Fja vraca "prom" promenljivu

```
44 def print_table(stanje: list[list], x: int, y: int):
45     pom=""
46     pom2=""
47     for i in range(x):
48         pom+=chr(65+i)+" "
49         pom2+="="
50
51     print(pom)
52     print(pom2)
53     pom3=str(y)+"||"
54     z = y
55     for w in stanje:
56         for i in range(0,len(w)):
57             if(i!=len(w)-1):
58                 pom3+=w[i]+"|"
59             else:
60                 pom3+=w[i]+"||"+str(z)
61         print(pom3)
62         z-=1
63         pom3=str(z)+"||"
64     print(pom2)
65     print(pom)
```

print_table(tabela,x,y) - prosledjuje se tabela, sirina i visina table. U prvoj for petlji koja se krece od 0 do sirine table u promenljivoj pom se ispisuju slova kao oznake kolona pocevsi od A, istovremeno se u pom2 upisuje oznaka = koja odvaja oznaku od polja u tabeli. Dalje formiranje table pocinje sa oznakom vrste gde se prvo upisuje broj vrste koji se zatim odvaja znakom "||" od celija table. U promenljivoj z se cuva broj vrsta sto je ujedno i redni broj prve vrste (oznaka prve vrste). Ova promenljiva se kasnije umanjuje i koristi za oznacavanje svake od vrsta.

for w in stanje: - U w izdvaja svaku od kolona na tabeli.

for i in range(0, len(w)): - I je indeks polja u okviru te vrste (klasicna for petlja za izdvajanje indeksa koja ide od 0 do broja elemenata).

if(i!=len(w)-1): - Proverava da li je rec o poslednjem polju u vrsti, u petlju se ulazi kad nije i polja se odvajaju sa "|". Kad prethodna for petlja dodje do zadnjeg elementa to je znak da se stampa ' "||"+str(z) ' z je i dalje oznaka vrste i dodaje se sa desne strane vrste u prikaz table.

Stampa se vrsta kao print(pom3)

z se smanjuje za 1 sto je priprema za obradu sledece vrste koja se opet priprema za stampu tako sto se unese oznaka vrste i znak "||". Nakon sto for petlja završi sa prolazom kroz sve vrste stampa se kraj prikaza.

```
41 def zapocniIgru():
42     print_table(tabla, valX, valY)
```

zapocniIgru()

Poziva samo fju za prikaz tabele jer se na pocetku igre samo stampa prazna tabela.

```
88 def igra():
89     zapocniIgru()
90     if(prvi=='X'):
91         igr = X
92     else:
93         igr = 0
94     while(proveriKraj(igr, tabla, valX, valY)>0):
95         if(odigrajPotez(igr)==True):
96             if(igr == X):
97                 igr = 0
98             else:
99                 igr = X
100
```

igra()

Fja igra obuhvata celu logiku igre. Prvo se bira koji igrac igra prvi u promenljivu "prvi" gde se u promenljivu "igr" cuva ta odluka, a data promenljiva ce dalje da se koristi da cuva trenutnog igraca na potezu.

while petlja proverava da li se doslo do kraja igre (fja proverikraj vraca broj preostalih poteza za igraca koji je na potezu. U trenutku kada taj broj dostigne 0 igra je završena i iz petlje se izlazi) unutar ove petlje se poziva fja

odigrajPotez koja treba da vrati "True" ako je odigran validan potez sto je znak da je sad red na drugog igraca da igra pa se promenljiva igr u sledecoj proveru promeni na oznaku drugog igraca (sa X na O i obrnuto).