

Извештај пројекта Диминација (Domineering)

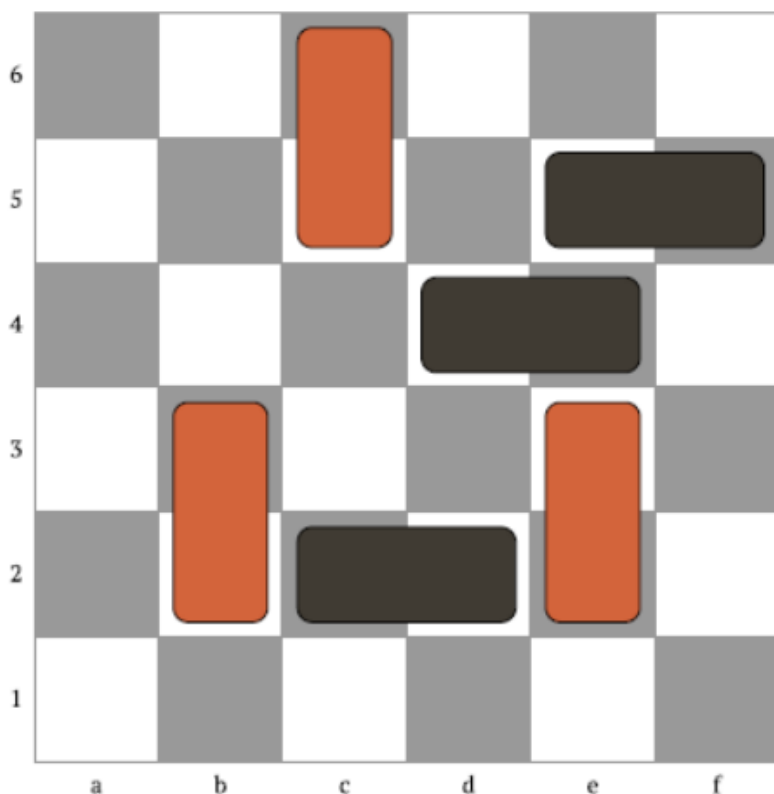
Никола Гоцић 16039

Аница Јоцић 16134

Јована Голијанин 16038

Опис проблема

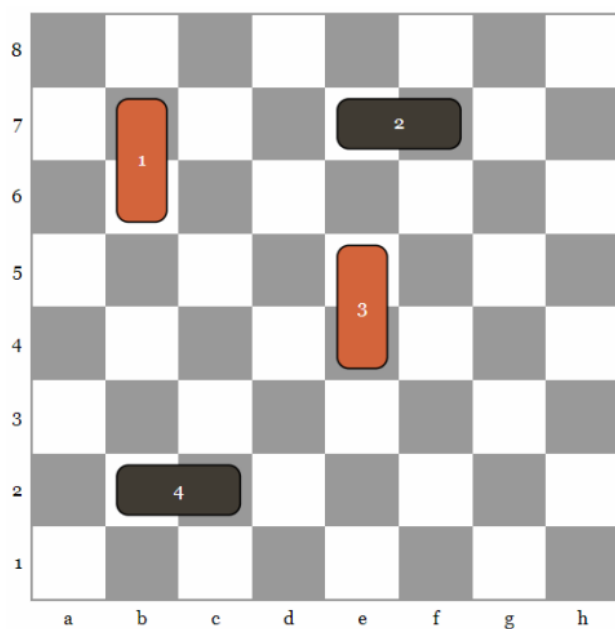
Проблем је стратешка игра постављања плочица на таблу која се зове Доминација (Domineering). Табла је произвољних димензија $m \times n$, m врста и n колона (препоручено је 8×8). Плочице су величине 2×1 . Два играча црни и бели (X и O) наизменично одигравају по један потез. Не постоји ограничење у броју плочица које играчи поседују. Један играч поставља плочице хоризонтално, док их други поставља вертикално. Поражен је онај играч који нема могућност да постави плочицу на таблу. Игра човек против рачунара и могуће изабрати да први игра човек или рачунар



Правила игре

На почетку игре табла је празна. Играчи повлаче потезе наизменично. Један играч поставља плочице само хоризонтално, док их други поставља само вертикално. Први играч поставља плочице вертикално. Играч, у једном потезу, поставља само једну плочицу на таблу у одговарајућем смеру (хоризонтално и вертикално). Плочице се не смеју преклапати, нити вирити ван табле.

Плочице се не могу поставити тако да почињу на пола поља и завршавати се на пола поља. Поражен је онај играч који нема могућност да постави плочицу на таблу.



	A	B	C	D	E	F	G	H	
	=	=	=	=	=	=	=	=	
8									8
	-	-	-	-	-	-	-	-	
7		X		O	O				7
	-	-	-	-	-	-	-	-	
6		X							6
	-	-	-	-	-	-	-	-	
5					X				5
	-	-	-	-	-	-	-	-	
4					X				4
	-	-	-	-	-	-	-	-	
3							X		3
	-	-	-	-	-	-	-	-	
2		O	O				X		2
	-	-	-	-	-	-	-	-	
1									1
	=	=	=	=	=	=	=	=	
	A	B	C	D	E	F	G	H	

Лоши потези X

1. [9 F]
2. [3 K]
3. [8 B]
4. [8 G]
3. [5 E]
4. [6 F]

Добри потези X

1. [1 A]
2. [2 E]
3. [5 H]
4. [7 D]

Фаза I - Формулација проблема и интерфејс

Потребно је дефинисати начин за представљање стања проблема (игре): табла, позиције плочица на табли, написати функције за постављање почетног стања које се дефинише на основу задате величине табле, написати функције за проверу краја игре (играч не може поставити ни једну плочицу на таблу) и написати функције које проверавају исправност унетог потеза.

Потребно је омогућити избор **ко ће играти први** (човек или рачунар).

Имплементирали смо функцију `koIgraPrvi()` која од корисника оцекује унос путем конзоле: 0 уколико човек игра први или 1 уколико рачунар игра први.

```
#Омогућити избор ко ће играти први (човек или рачунар)
def koIgraPrvi():
    global igraPrvi

    print("Ko igra prvi? (0-covek, 1-racunar):")
    igraPrvi = int(sys.stdin.readline())
    if(igraPrvi==0):
        print("Covek je prvi na potezu!")
    else:
        print("Racunar je prvi na potezu!")
```

Потребно је имплементирати функције које обезбеђују **унос почетних параметара игре**. Унос димензија табле (m и n),

као и функције које обезбеђују **прављење иницијалног стања проблема** (игре) тј. прављење празне табле на основу задатих димензија (m и n).

Имплементирали смо функцију `init()` која обезбеђују унос почетних параметара игре тј. димензије табле (m и n) које корисник уноси путем конзоле и прављење празне табле на основу задатих димензија (m и n) која је представљена матрицом

```
#Implementirati funkcije koje obezbeđuju unos početnih parametara igre
def init():
    global matrica
    global brojVrsta
    global brojKolona

    print("Unesite broj vrsta:")
    brojVrsta = int(sys.stdin.readline())

    print("Unesite broj kolona:")
    brojKolona = int(sys.stdin.readline())

    matrica = [ [ " " for i in range(brojKolona) ] for j in range(brojVrsta) ]
```

Потребно је имплементирати функције које обезбеђују **приказ произвољног стања** проблема (игре) као и приказ задате табле.

Имплементирали смо функције `stampaj()`, `top(brojKolona)` и `bottom(brojKolona)` које врше приказ произвољног стања и задате табле.

```

#Implementirati funkcije koje obezbeđuju prikaz proizvoljnog stanja problema (igre)
def stampaj():
    top(brojKolona)

    for x in range(0,brojVrsta):
        print(str(brojVrsta-x)+"||",end='')
        for num in range(0,brojKolona):
            print(matrica[x][num]+'|',end='')
        print("||"+str(brojVrsta-x))

    bottom(brojKolona)

def top(brojKolona):
    print(' ',end='')
    for num in range(0,brojKolona):
        print(' '+string.ascii_uppercase[num],end='')
    print('')
    print(' ',end='')
    for num in range(0,brojKolona):
        print(' =',end='')
    print('')

def bottom(brojKolona):
    print(' ',end='')
    for num in range(0,brojKolona):
        print(' =',end='')
    print('')
    print(' ',end='')
    for num in range(0,brojKolona):
        print(' '+string.ascii_uppercase[num],end='')
    print('')

```

Потребно је реализовати функције за **унос потеза**. Потез је облика играч (X или O) и позиција доњег левог поља где ће се наћи плочица [4,C]. Ако је играч X (вертикално), плочица би требало да се постави на поља [4,C] и [5,C]. Ако је играч O (хоризонтално), плочица би требало да се постави на поља [4,C] и [4,D].

Имплементирали смо функцију `odigrajPotez(vrsta,kolona)` која проверава да ли је потез ваљан и уколико јесте,

проверава да ли је играч X или играч O на потезу; одиграва потез за одговарајућег играча тиме што на таблу поставља одговарајући знак (вертикалну домину за играча X или хоризонталну домину за играча O); након одиграног потеза штампа одиграни потез. Такође, проверава да ли смо дошли до краја игре или смо унели невалидан потез.

```
| | print("Uneli ste nevalidni potez!")
```

Потребно је реализовати функције које проверавају **да ли је потез ваљан**. Проверити да плочица није ван табле. Проверити да се плочица не поклапа са неком већ постављеном плочицом.

Имплементирали смо функцију `potezValjan(vrsta,kolona)` која проверава да ли је плочица ван табле и да ли се плочица не поклапа са неком већ постављеном плочицом.

```

#Realizovati funkcije koje proveravaju da li je potez valjan
# vrsta 1..N | kolona A..Z
def potezValjan(vrsta,kolona):
    kol = ord(kolona)-65
    vrs = brojVrsta-vrsta

    if(naPotezu=='X'):
        if(vrs+1>=brojVrsta or kol>=brojKolona):
            return False

        if(matrica[vrs][kol]==' ' and matrica[vrs+1][kol]==' '):
            return True
        else:
            return False
    else:
        if(vrs>=brojVrsta or kol+1>=brojKolona):
            return False

        if(matrica[vrs][kol]==' ' and matrica[vrs][kol+1]==' '):
            return True
        else:
            return False

```

Потребно је написати функције за **проверу краја игре** који наступа уколико играч не може поставити ни једну плочицу на таблу.

Имплементирали смо функцију `krajIgre()` која проверава да ли постоје слободна поља за играча који је на потезу. Уколико нема слободних поља тада је крај игре.


```

#Napisati funkcije za proveru kraja igre
def krajIgre():
    if(naPotezu=='X'):
        for i in range(0,brojVrsta-1):
            for j in range(0,brojKolona):
                if(matrica[i][j]==' ' and matrica[i+1][j]==' '):
                    return False
    else:
        for i in range(0,brojVrsta):
            for j in range(0,brojKolona-1):
                if(matrica[i][j]==' ' and matrica[i][j+1]==' '):
                    return False
    return True #Kraj je igre zato sto nema slobodnih polja za igraca koj je na potezu

```

На крају смо имплементирали функцију `start()` која позива горе дефинисане функције; у почетку испитује који је играч први на потезу, поставља иницијално стање и штампа почетни изглед табле. Докле год играчи имају слободних места где могу да поставе своју домину, они наизменично одигравају потезе. Када више нема слободних места ни за једног играча, игра је готова. Победио је онај играч који је последњи поставио домину на таблу а изгубио онај који је први остао без могућности да постави своју домину.

```

def start():
    global kraj

    koIgraPrvi()

    init()

    stampaj()

    while True:
        if(krajIgre()):
            pobedio=""
            if(naPotezu=='X'):
                pobedio='O'
            else:
                pobedio='X'
            print("Kraj Igre, pobedio je igrac "+pobedio+"!!!")
            break

        print("\nIgrac "+naPotezu+" na potezu")
        print("Unesite vrstu (broj 1..N):")
        vrsta = int(sys.stdin.readline())
        if vrsta == -1:
            break
        print("Unesite kolonu (slovo A..Z):")
        kolona = sys.stdin.readline()
        odigrajPotez(vrsta,kolona[0])

start()

```

Пример једног позива функције start():

Ko igra prvi? (0-covek, 1-racunar):

0

Covek je prvi na potezu!

Unesite broj vrsta:

6

Unesite broj kolona:

6

	A	B	C	D	E	F	
	=	=	=	=	=	=	
6							6
5							5
4							4
3							3
2							2
1							1
	=	=	=	=	=	=	
	A	B	C	D	E	F	

Igrac X na potezu

Unesite vrstu (broj 1..N):

6

Unesite kolonu (slovo A..Z):

A

	A	B	C	D	E	F	
	=	=	=	=	=	=	
6	X						6
5	X						5
4							4
3							3
2							2
1							1
	=	=	=	=	=	=	
	A	B	C	D	E	F	

Igrac 0 na potezu
Unesite vrstu (broj 1..N):
4
Unesite kolonu (slovo A..Z):
A

	A	B	C	D	E	F	
6	X						6
5	X						5
4	0 0						4
3							3
2							2
1							1

Igrac X na potezu
Unesite vrstu (broj 1..N):
3
Unesite kolonu (slovo A..Z):
A

	A	B	C	D	E	F	
6	X						6
5	X						5
4	0 0						4
3	X						3
2	X						2
1							1

```

Igrac 0 na potezu
Unesite vrstu (broj 1..N):
1
Unesite kolonu (slovo A..Z):
A
  A B C D E F
  = = = = =
6||X| | | | |6
5||X| | | | |5
4||0|0| | | |4
3||X| | | | |3
2||X| | | | |2
1||0|0| | | |1
  = = = = =
  A B C D E F

```

```

Igrac X na potezu
Unesite vrstu (broj 1..N):
5
Unesite kolonu (slovo A..Z):
D
  A B C D E F
  = = = = =
6||X| | | | |6
5||X| |X| | |5
4||0|0|X| | |4
3||X| | | | |3
2||X| | | | |2
1||0|0| | | |1
  = = = = =
  A B C D E F

```

Igrac 0 na potezu
Unesite vrstu (broj 1..N):
1
Unesite kolonu (slovo A..Z):
C

	A	B	C	D	E	F	
6	X						6
5	X		X				5
4	O	O	X				4
3	X						3
2	X						2
1	O	O	O	O			1

Igrac X na potezu
Unesite vrstu (broj 1..N):
6
Unesite kolonu (slovo A..Z):
C

	A	B	C	D	E	F	
6	X	X					6
5	X	X	X				5
4	O	O	X				4
3	X						3
2	X						2
1	O	O	O	O			1

Igrac 0 na potezu
Unesite vrstu (broj 1..N):
3
Unesite kolonu (slovo A..Z):
B

	A	B	C	D	E	F	
	=	=	=	=	=	=	
6	X	X					6
5	X	X X					5
4	0 0	X					4
3	X 0 0						3
2	X						2
1	0 0 0 0						1
	=	=	=	=	=	=	
	A	B	C	D	E	F	

Igrac X na potezu
Unesite vrstu (broj 1..N):
6
Unesite kolonu (slovo A..Z):
E

	A	B	C	D	E	F	
	=	=	=	=	=	=	
6	X	X	X				6
5	X	X X X					5
4	0 0	X					4
3	X 0 0						3
2	X						2
1	0 0 0 0						1
	=	=	=	=	=	=	
	A	B	C	D	E	F	

Igrac 0 na potezu
Unesite vrstu (broj 1..N):
2
Unesite kolonu (slovo A..Z):
C

	A	B	C	D	E	F	
6	X	X	X				6
5	X	X	X X				5
4	0	0	X				4
3	X	0 0					3
2	X	0 0					2
1	0	0 0 0					1
	A	B	C	D	E	F	

Igrac X na potezu
Unesite vrstu (broj 1..N):
6
Unesite kolonu (slovo A..Z):
F

	A	B	C	D	E	F	
6	X	X	X	X			6
5	X	X	X X	X X			5
4	0	0	X				4
3	X	0 0					3
2	X	0 0					2
1	0	0 0 0					1
	A	B	C	D	E	F	

Igrac O na potezu
Unesite vrstu (broj 1..N):
3
Unesite kolonu (slovo A..Z):
E

	A	B	C	D	E	F	
	=	=	=	=	=	=	
6	X	X	X	X		6	
5	X	X	X	X	X	5	
4	0	0	X			4	
3	X	0	0	0	0	3	
2	X	0	0			2	
1	0	0	0	0		1	
	=	=	=	=	=	=	
	A	B	C	D	E	F	

Igrac X na potezu
Unesite vrstu (broj 1..N):
6
Unesite kolonu (slovo A..Z):
B

	A	B	C	D	E	F	
	=	=	=	=	=	=	
6	X	X	X	X	X	6	
5	X	X	X	X	X	5	
4	0	0	X			4	
3	X	0	0	0	0	3	
2	X	0	0			2	
1	0	0	0	0		1	
	=	=	=	=	=	=	
	A	B	C	D	E	F	

Igrac 0 na potezu

Unesite vrstu (broj 1..N):

2

Unesite kolonu (slovo A..Z):

E

A B C D E F

= = = = = =

6||X|X|X| |X|X||6

5||X|X|X|X|X|X||5

4||0|0| |X| | ||4

3||X|0|0| |0|0||3

2||X| |0|0|0|0||2

1||0|0|0|0| | ||1

= = = = = =

A B C D E F

Kraj Igre, pobedio je igrac 0!!!

PS C:\Vestacka\Projekat> |