



ALGORITMI IN PODATKOVNE STRUKTURE 1

2. laboratorijske vaje

Rekurzija (težje naloge)

NALOGA 1

Napišite funkcijo, ki izpiše vse pravilne oklepajske izraze, sestavljene iz N znakov '(' in N znakov ') '.

Oklepajski izraz je pravilen, če so oklepaji pravilno gnezdeni.

Pravilni oklepajski izrazi za $N=3$ so:

((()))
(())
()) (
() (())
() () ()

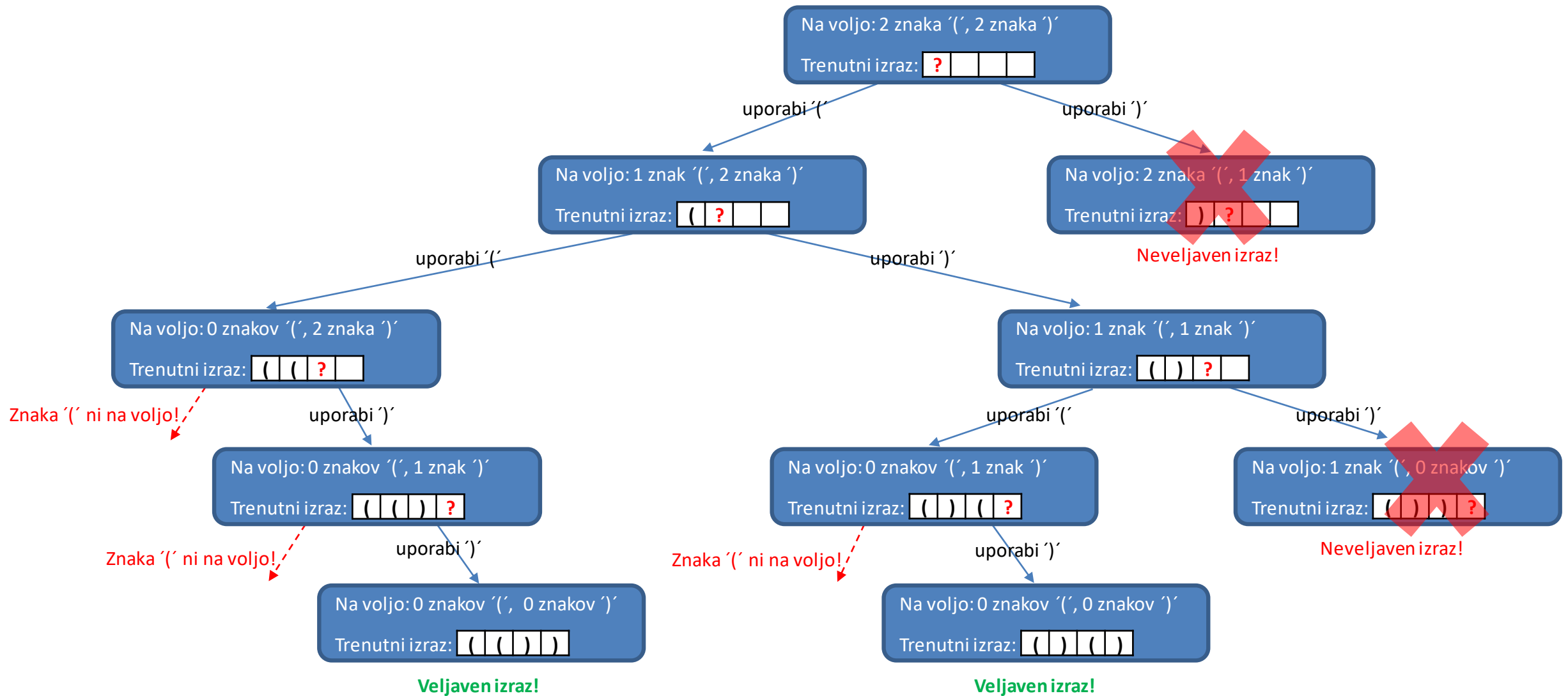
Nekaj primerov nepravilnih oklepajskih izrazov:

) () ((
()) ((
(()) ((

Ideja: izraz gradimo sproti in preizkušamo vsa možna nadaljevanja, pri katerih število zaklepajev ne presega števila oklepajev (gledano z leve proti desni).

NALOGA 1

Primer za N=2



NALOGA 1

Reši problem do konca, če je ostalo X znakov '(' in Y znakov ')'

Robni pogoj:
Če velja $Y=0$, je izraz sestavljen do konca.

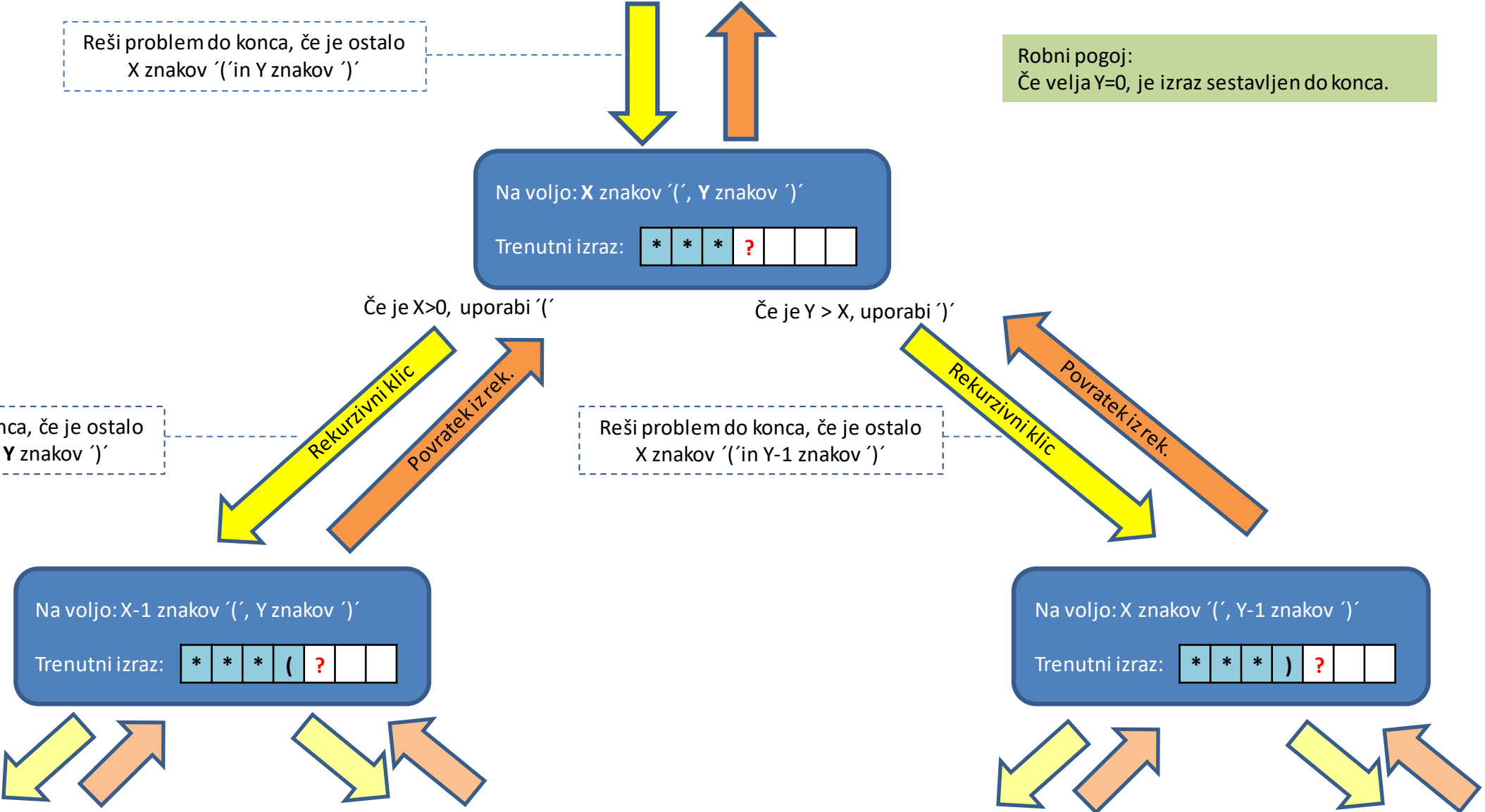
Na voljo: X znakov '(' , Y znakov ')'
Trenutni izraz: * * * ?

Reši problem do konca, če je ostalo X-1 znakov '(' in Y znakov ')'

Reši problem do konca, če je ostalo X znakov '(' in Y-1 znakov ')'

Na voljo: X-1 znakov '(' , Y znakov ')'
Trenutni izraz: * * * (?


Na voljo: X znakov '(' , Y-1 znakov ')'
Trenutni izraz: * * *) ?



NALOGA 1

```
public static void sestaviIzraz(int N)
{
    char[] izraz = new char[2*N];
    nadaljujIzraz(N, N, izraz, 0);
}
```

št. '(' št. ')' pozicija v polju "izraz", v katero pišemo



```
public static void nadaljujIzraz(int X, int Y, char[] izraz, int n)
{
    //robni pogoj, ko nimamo na voljo nobenega znaka ')'
    if (Y == 0)
    {
        System.out.println(izraz);
        return;
    }

    //lahko izraz nadaljujemo z '(', če jih še imamo na voljo
    if (X > 0)
    {
        izraz[n] = '(';
        nadaljujIzraz(X-1, Y, izraz, n+1);
    }

    //lahko izraz nadaljujemo z ')', če s tem ne bomo presegli števila uporabljenih '('
    if (Y > X)
    {
        izraz[n] = ')';
        nadaljujIzraz(X, Y-1, izraz, n+1);
    }
}
```

NALOGA 2



Podano je polje števil in vrednost N . Napišite funkcijo, ki izbere (ne nujno najmanjšo) podmnožico elementov tega polja tako, da je vsota le-teh enaka N .

Na primer, za polje $\{7, 8, 5, 4, 9, 2, 5\}$ in $N=10$, zahtevano vsoto dosežemo z izbiro elementov 8 in 2.

Ideja: Če izberemo prvi element, recimo z vrednostjo v , potem iz preostalih elementov moramo sestaviti znesek $N-v$. Če nam to ne uspe, je potrebno celotni znesek N sestaviti brez uporabe prvega elementa.

NALOGA 2



polje:

v_1	v_2	...	v_m
-------	-------	-----	-------

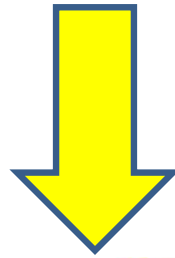
Robni pogoji:	
znesek: 0	✓
znesek: < 0	✗
znesek: > 0 polje: \emptyset	✗

NALOGA 2

Ali lahko sestaviš znesek **N**
z elementi polja

v	...ostali elementi...
---	-----------------------

 ?



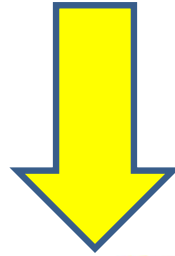
Kako naj rešim nalogo, če
robni pogoji niso izpolnjeni?

NALOGA 2

Ali lahko sestaviš znesek **N**
z elementi polja

v	...ostali elementi...
---	-----------------------

 ?



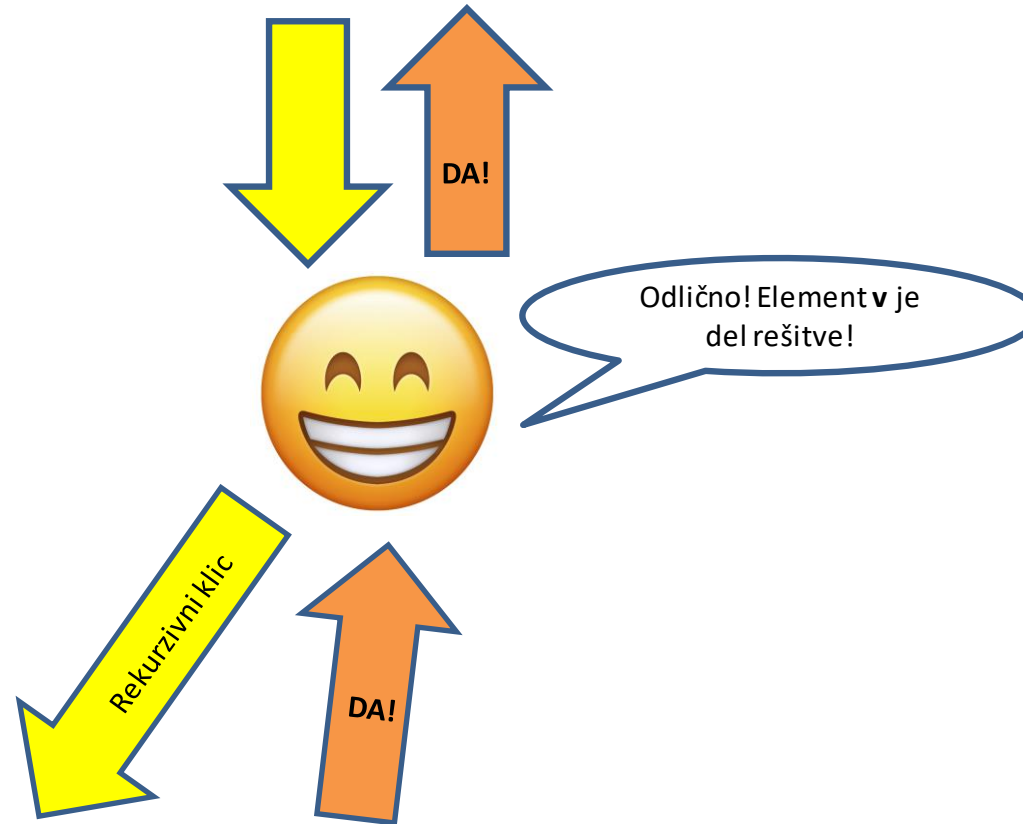
Ali lahko sestaviš znesek **N-v**
z elementi polja

...ostali elementi...

 ?

NALOGA 2

Ali lahko sestaviš znesek **N**
z elementi polja **v** ...ostali elementi... ?



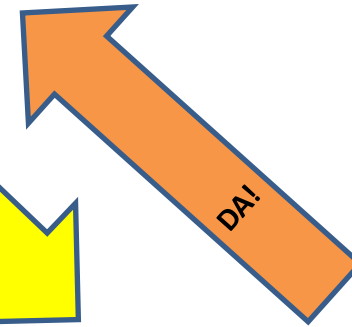
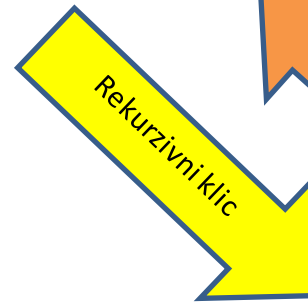
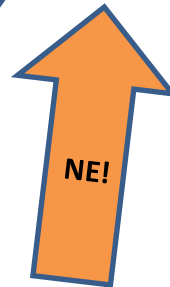
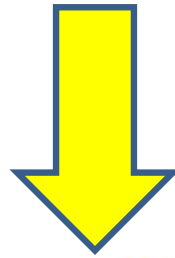
Ali lahko sestaviš znesek **N-v**
z elementi polja ...ostali elementi... ?

NALOGA 2

Ali lahko sestaviš znesek **N**
z elementi polja

v	...ostali elementi...
---	-----------------------

 ?



Ali lahko sestaviš znesek **N-v**
z elementi polja

...ostali elementi...

 ?

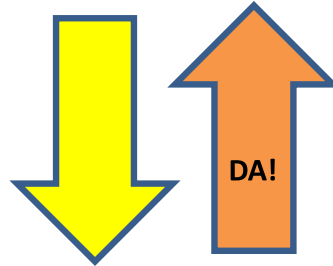
Ali lahko sestaviš znesek **N**
z elementi polja

...ostali elementi...

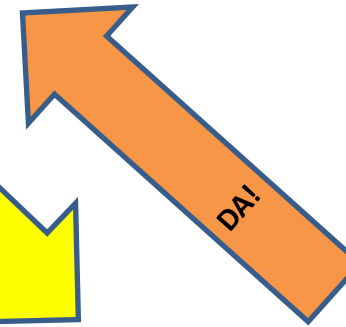
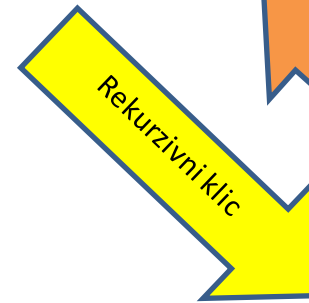
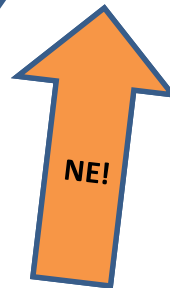
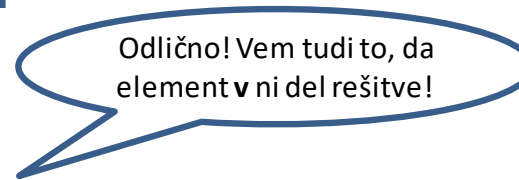
 ?

NALOGA 2

Ali lahko sestaviš znesek **N**
z elementi polja ?



DA!



Ali lahko sestaviš znesek **N-v**
z elementi polja ?

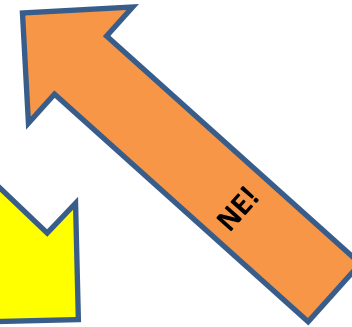
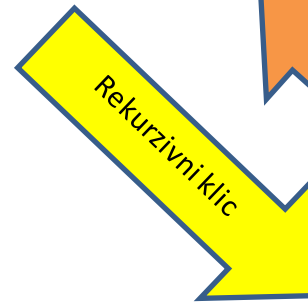
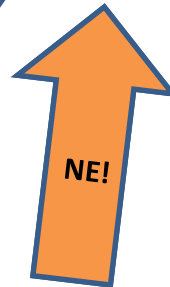
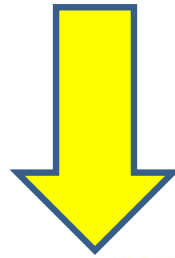
Ali lahko sestaviš znesek **N**
z elementi polja ?

NALOGA 2

Ali lahko sestaviš znesek **N**
z elementi polja

v	...ostali elementi...
---	-----------------------

 ?



Ali lahko sestaviš znesek **N-v**
z elementi polja

...ostali elementi...

 ?

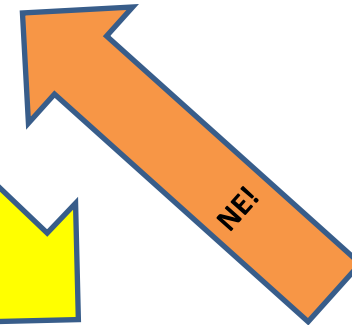
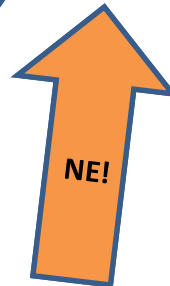
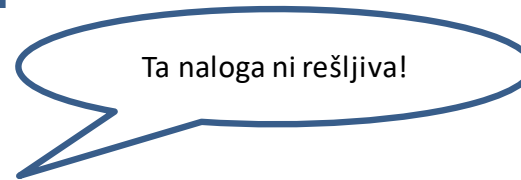
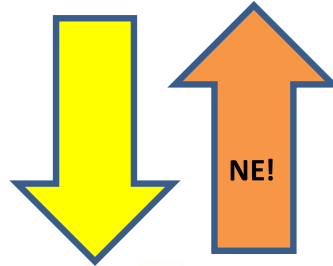
Ali lahko sestaviš znesek **N**
z elementi polja

...ostali elementi...

 ?

NALOGA 2

Ali lahko sestaviš znesek **N**
z elementi polja ?

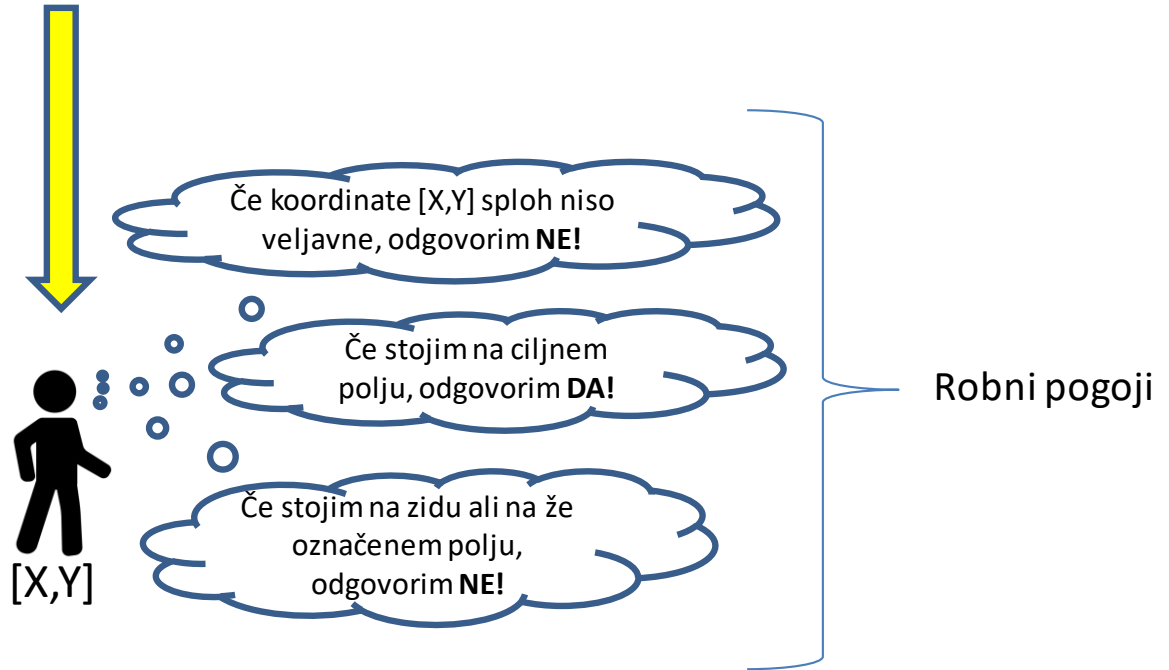


Ali lahko sestaviš znesek **N-v**
z elementi polja ?

Ali lahko sestaviš znesek **N**
z elementi polja ?

NALOGA 3

Ali je polje $[X,Y]$ na poti do izhoda?

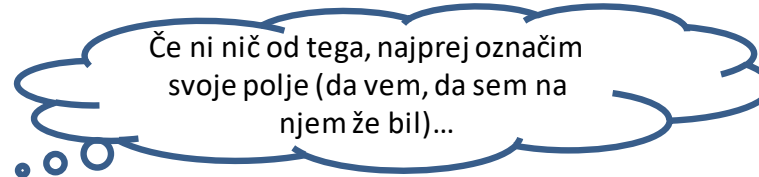


NALOGA 3

Ali je polje [X,Y] na poti do izhoda?



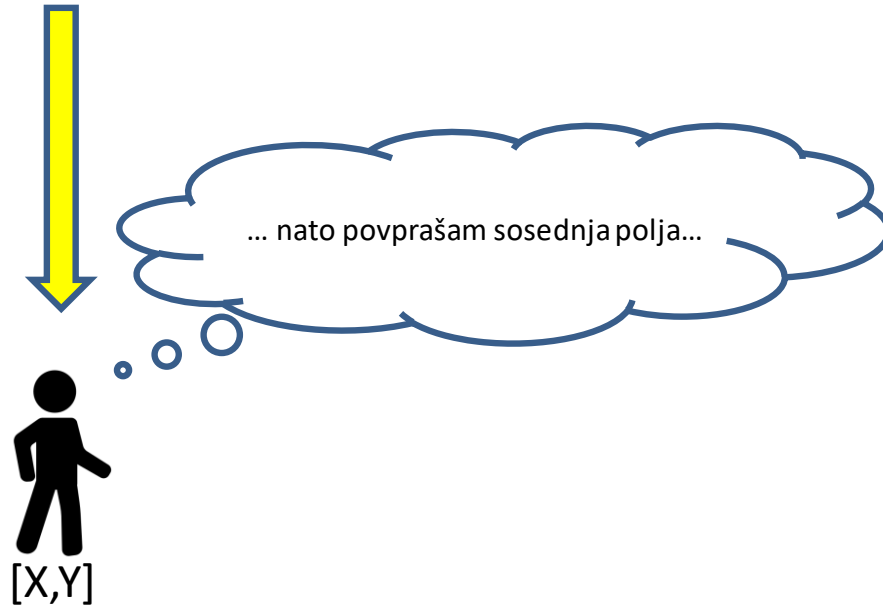
[X,Y]



tako preprečimo vrtenje v krogu!

NALOGA 3

Ali je polje [X,Y] na poti do izhoda?

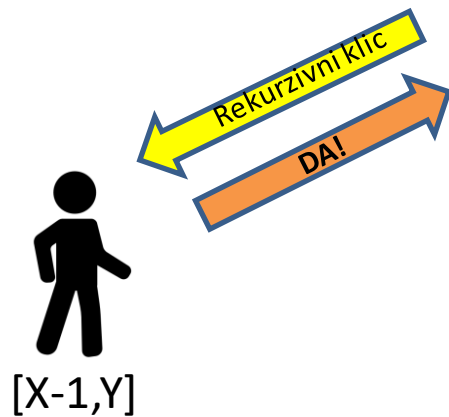


NALOGA 3

Ali je polje $[X,Y]$ na poti do izhoda?

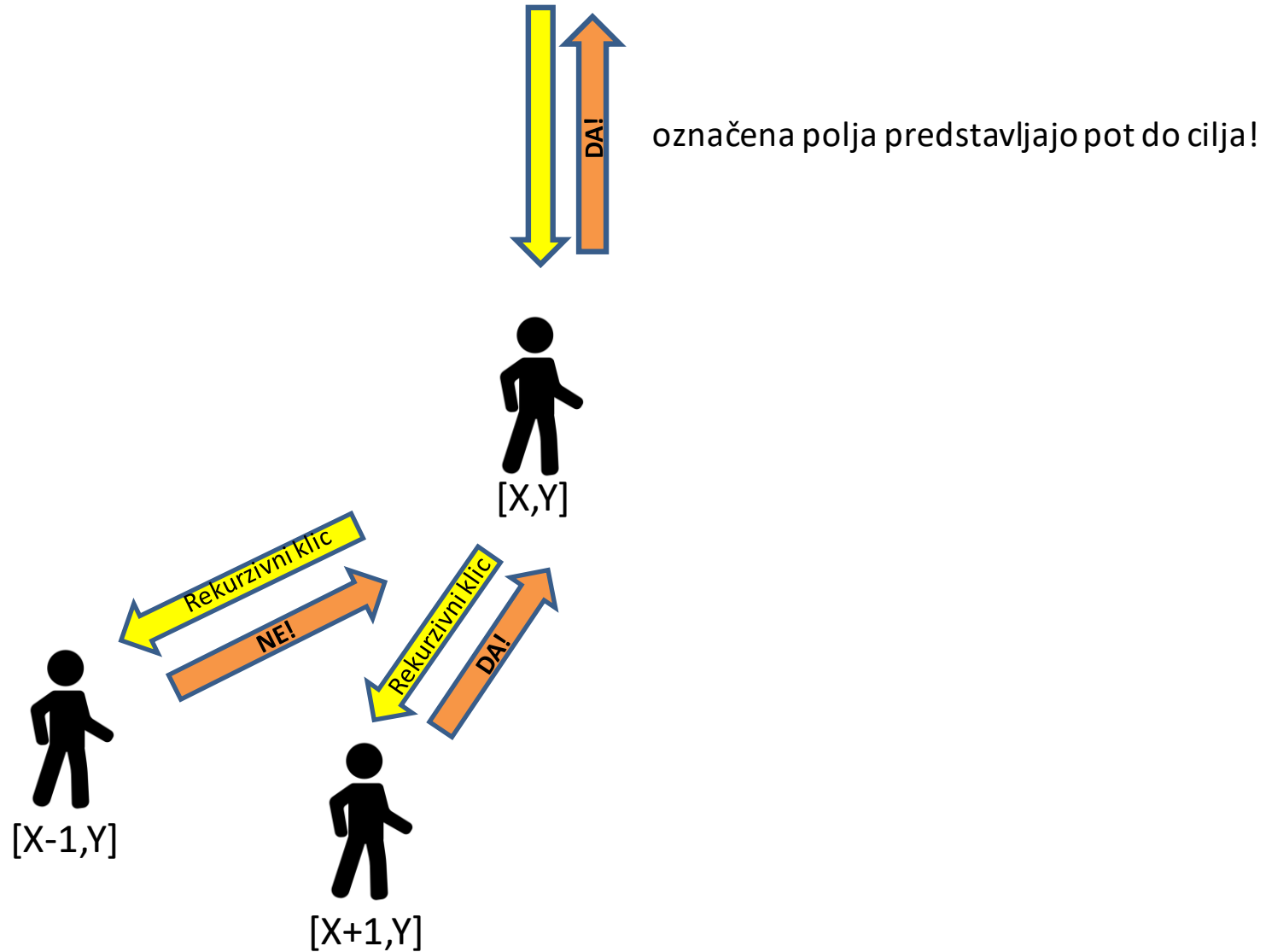


označena polja predstavljajo pot do cilja!



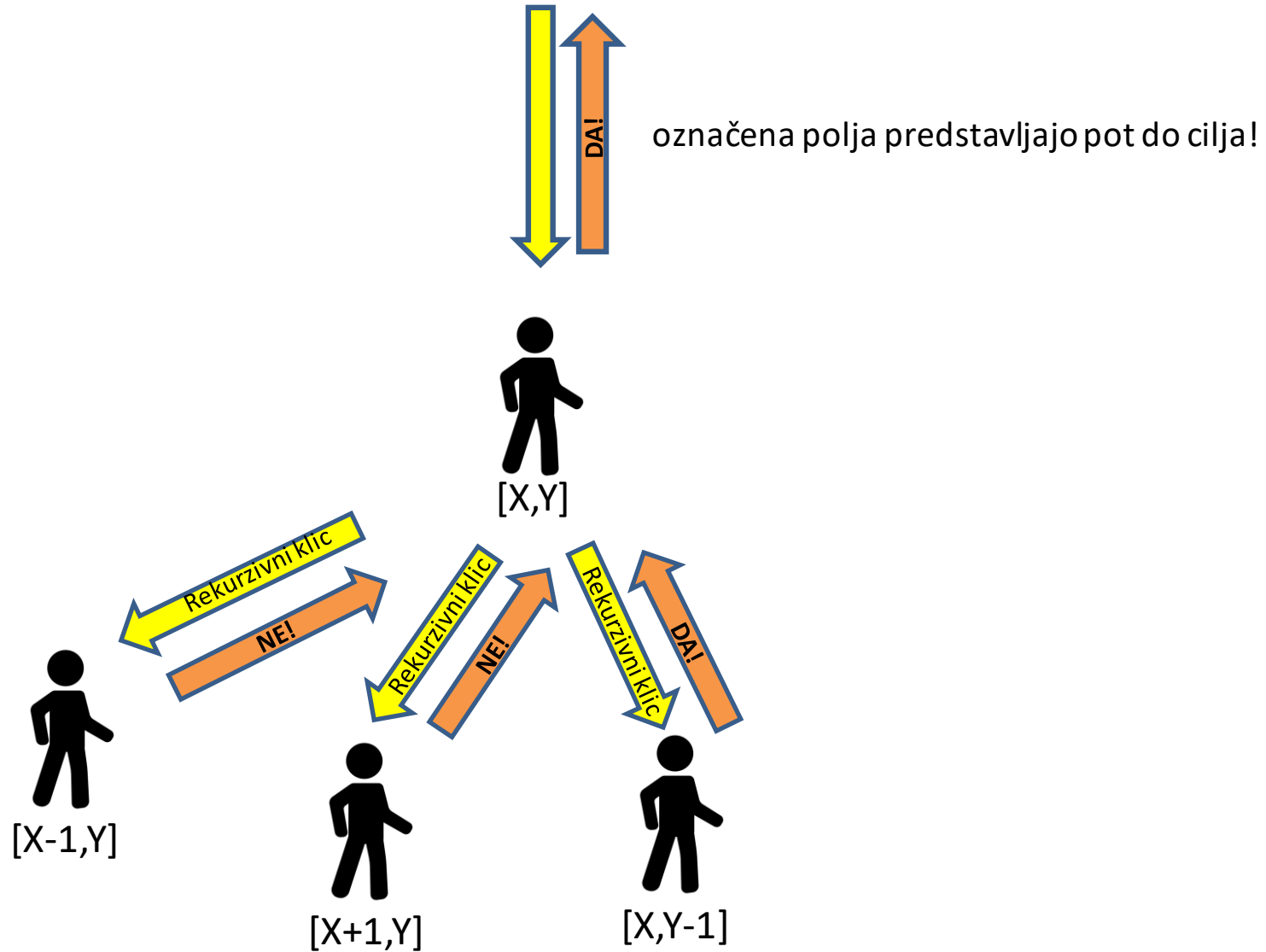
NALOGA 3

Ali je polje [X,Y] na poti do izhoda?



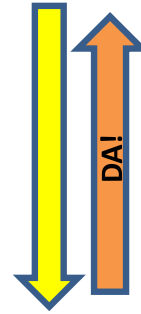
NALOGA 3

Ali je polje $[X,Y]$ na poti do izhoda?

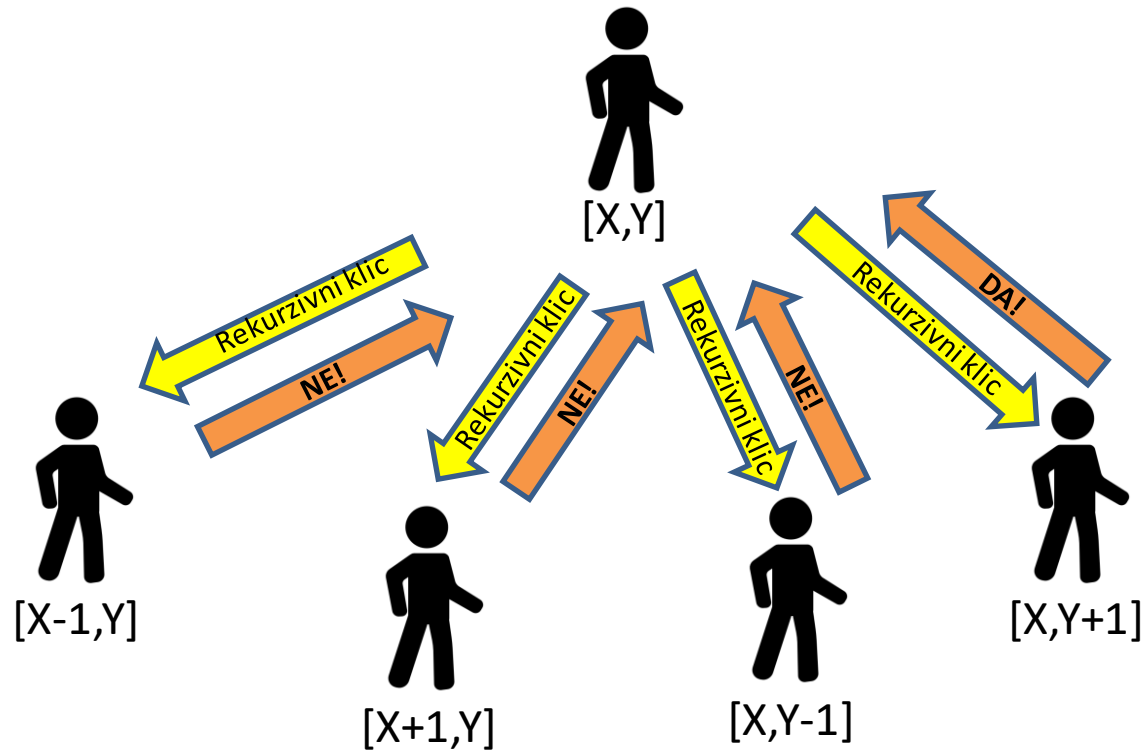


NALOGA 3

Ali je polje $[X,Y]$ na poti do izhoda?

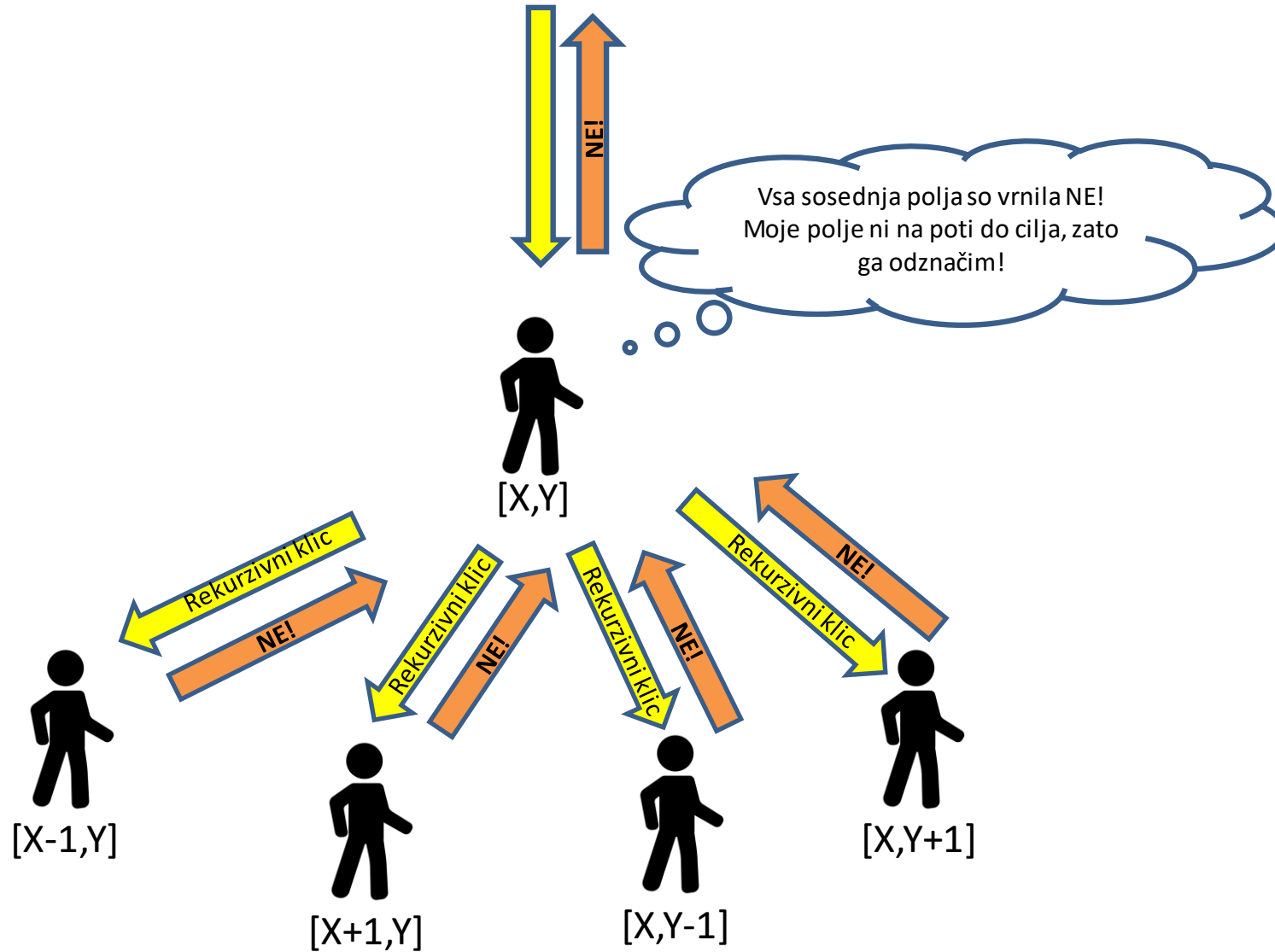


označena polja predstavljajo pot do cilja!



NALOGA 3

Ali je polje $[X,Y]$ na poti do izhoda?



NALOGA 4

Napišite funkcijo, ki reši podano sudoku uganko.

Pravila:

- cilj je zapolniti kvadratno mrežo s števili od 1 do 9
- vsako število se lahko pojavi točno enkrat v vsakem stolpcu, vsaki vrstici in vsakem manjšem kvadratu velikosti 3×3
- v mreži so nekatera števila že podana

Pri reševanju uporabite strategijo poskušanja in vračanja.

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

NALOGA 4

- rešujemo vrstico po vrstico; po posamezni vrstici se premikamo po poljih z leve proti desni

- če je polje izpolnjeno, rekurzivno rešimo sosednje polje in vrnemo dobljeni rezultat

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

- če je polje prazno:

- izberemo dovoljeno vrednost med 1 in 9 (glede na trenutne vrednosti v mreži) ter jo vpišemo v polje
- rekurzivno rešimo sosednje polje
- če rekurzija uspe, vrnemo pozitiven rezultat
- če rekurzija ne uspe, izberemo drugo vrednost, jo vpišemo v polje in ponovno poskusimo rešiti sosednje polje
- če za nobeno vrednost rekurzija ne uspe, izbrišemo polje in vrnemo negativen rezultat

- robna primera:

- če gremo čez desni rob zadnje vrstice, vrnemo pozitiven rezultat
- če reševanje prvega polja v prvi vrstici vrne negativen rezultat, naloga ni rešljiva

NALOGA 4

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

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

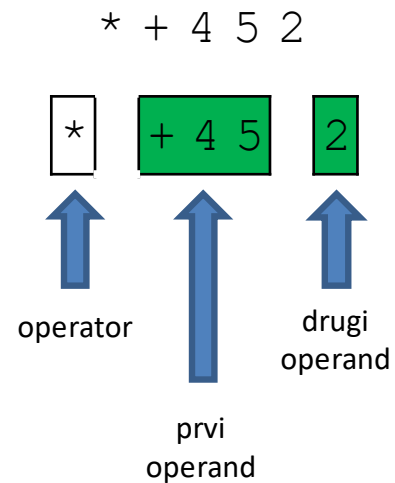
- rešujemo vrstico po vrstico; po posamezni vrstici se premikamo po poljih z leve proti desni
- če je polje izpolnjeno, rekurzivno rešimo sosednje polje in vrnemo dobljeni rezultat
- če je polje prazno:
 - izberemo dovoljeno vrednost med 1 in 9 (glede na trenutne vrednosti v mreži) ter jo vpišemo v polje
 - rekurzivno rešimo sosednje polje
 - če rekurzija uspe, vrnemo pozitiven rezultat
 - če rekurzija ne uspe, izberemo drugo vrednost, jo vpišemo v polje in ponovno poskusimo rešiti sosednje polje
 - če za nobeno vrednost rekurzija ne uspe, izbrišemo polje in vrnemo negativen rezultat
- robna primera:
 - če gremo čez desni rob zadnje vrstice, vrnemo pozitiven rezultat
 - če reševanje prvega polja v prvi vrstici vrne negativen rezultat, naloga ni rešljiva

NALOGA 5

Napišite funkcijo, ki prejme izraz v prefiksni obliki in ga izpiše v infiksni obliki.

Za prefiksno notacijo velja, da se operatorji pišejo pred operandoma. Operandi so lahko konstante ali so tudi sami izrazi v prefiksni obliki.

Primer izraza v prefiksni obliki:

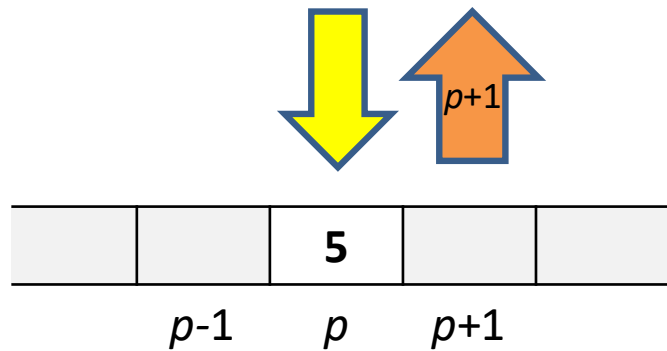


Isti izraz zapisan v infiksni obliki:

((4 + 5) * 2)

NALOGA 5

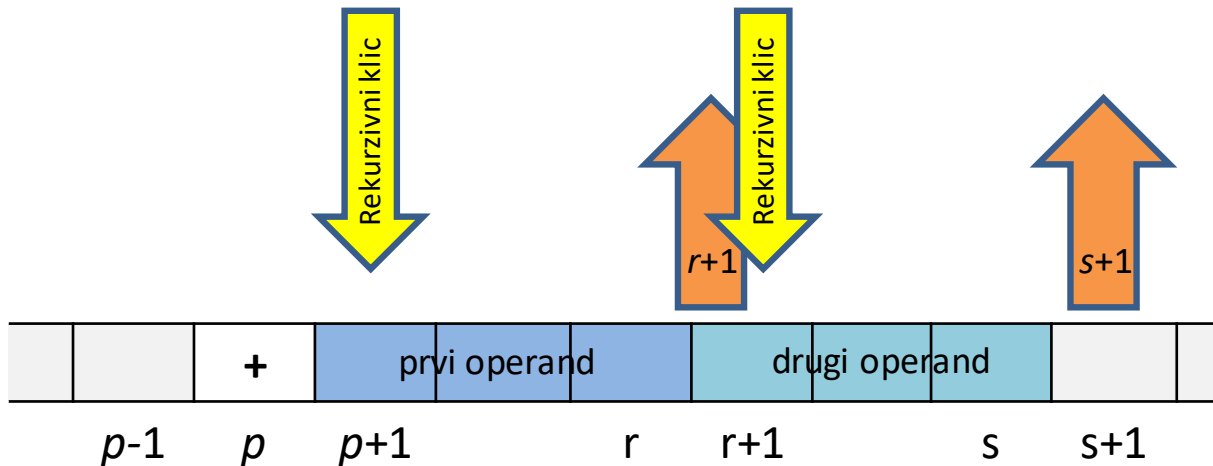
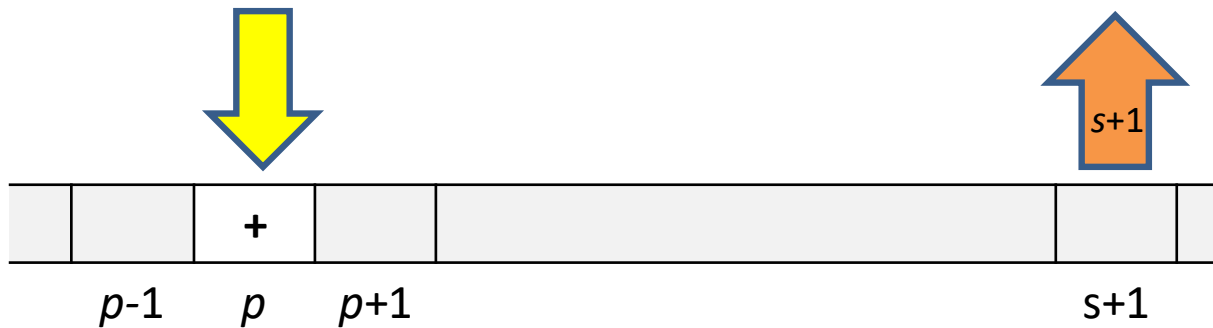
Izpiši izraz, ki začne na indeksu p



če se na poziciji p nahaja konstanta, jo izpišemo in vrnemo pozicijo $p+1$ (pozicija naslednjega neobdelanega elementa polja)

NALOGA 5

Izpiši izraz, ki začne na indeksu p



če se na poziciji p nahaja operacija (+, -, * ali /):

- izpišemo '('
- z rekurzijo izpišemo prvi operand
- izpišemo ')'
- izpišemo operacijo
- izpišemo '('
- z rekurzijo izpišemo drugi operand
- izpišemo ')'