

Logičko i funkcijsko programiranje

Primer drugog kolokvijuma

Na *Desktop*-u u direktorijumu *Rad* kreirati direktorijum *ImePrezime_BrIndeksa* i unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Nazivi fajlova **moraju** biti oblika *zadatak1.pl*, *zadatak2.pl*, *zadatak3.pl* i *zadatak4.pl*.

1.

- a. (3 poena) Написати пролошки предикат `ispis` који за дати број n исписује шему облика:

```
 $n$ 
 $n-1\ n$ 
 $n-1\ n-2\ n$ 
...
 $2\ 3\ \dots\ n$ 
 $1\ 2\ 3\ \dots\ n$ 

?- ispis(5).
5
4 5
3 4 5
2 3 4 5
1 2 3 4 5
true.
```

- b. (2 poena) Написати пролошки предикат `ispis2` који исписује речима дати број n .

```
?- ispis2(1234).
Jedan dva tri cetiri
true.
```

2. (6 poena) Написати пролошки предикат `max_cifra(Broj, Cifra)` који за дати цео број (`Broj`) одређује највећу цифру(`Cifra`).

```
?- max_cifra(1234, X).
X = 4.
```

3. (6 poena) Четири госпође се зову Ана, Марија, Рушка и Владислава а презивају Андрић, Петровић, Бранковић и Давидовић. Пошто организију новогодишњу прославу, договориле су се да свака донесе понешто.

- a. Госпођа Андрић ће донети чоколадну тарту
- b. Ни госпођа Бранковић ни Владислава неће донети колачиће
- c. Рушка, која није Давидовић, ће донети кафу
- d. Марија неће донети вино

Написати пролошки предикат којим се одређује како се која госпођа зове и презива као и шта ће која од њих донети на прославу.

```
?- pronadji([P,Q,R,S]).
P = [andric, marija, torta],
Q = [petrovic, ruska, kafa],
R = [brankovic, vladislava, vino],
S = [davidovic, ana, kolaci]
```

4. (6 poena) У поља матрице уписане су структуре облика `op(a,b)` где је a операција b цео број. Дозвољене операције су сабирање, одузимање и множење. Написати следеће пролошке предикате:

- a. `formTablu` којим се омогућава унос елемената матрице. Сваки елемент матрице је облика `op(znakOperacije, argument)`.

```
?- formTablu.  
Unesi dimenzije matrice 2.  
|: 2.  
Unesi elemente matrice  
Elementi matrice su operacije u obliku op(znak_operacije, argument)  
Unesi element na poziciji 1 1  
|: op(*,-1).  
Unesi element na poziciji 1 2  
|: op(+,10).  
Unesi element na poziciji 2 1  
|: op(-,3).  
Unesi element na poziciji 2 2  
|: op(*,5).  
:- dynamic mat/3.  
  
mat(1, 1, op(*, -1)).  
mat(1, 2, op(+, 10)).  
mat(2, 1, op(-, 3)).  
mat(2, 2, op(*, 5)).
```

- b. `nadjiSve` : којим се проналазе суме свих могућих путева кроз матрицу од доњег десног угла до горњег левог угла, при чему је дозвољено кретање у лево и на горе, и формира се и исписује листа која садржи пронађене суме.

```
?- nadjiSve.  
[50,-15]  
true.
```

- c. `nadjiMinimalno` : који, на основу формиране листе свих путева, проналази пут који има најмању суму и исписује одговарајућу путању кроз матрицу.

```
?- nadjiMinimalno.  
Min put ima sumu -15  
[mat(1,1,op(*,-1)),mat(2,1,op(-,3)),mat(2,2,op(*,5))]  
True.
```

Уграђени предикати који се могу користити :

- `member(X, L)` – True ако је X елемент листе L
- `append(A, B, C)` – True ако се спајањем листе A и листе B добија листа C
- `permutation(A, B)` – True ако је A пермутација листе B