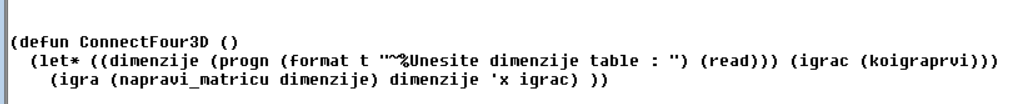
Nacin Predstavljanja Stanja igre:

((- - - - - - - - - - - - - - - -) (- - - - - - - - - - - - - - - -) (- - - - - - - - - - - - - - - -) (- - - - - - - - - - - - - - - -))

Svaka podlista predstavlja jednu tablu po Z osi, dok unutar podliste imamo tablu n\*n (u ovom slucaju 4\*4), gde na svakih n se nalazi jedna vrsta, prvih n predstavljaju n-ti red, drugih n predstavljaju (n-1)-ti itd.

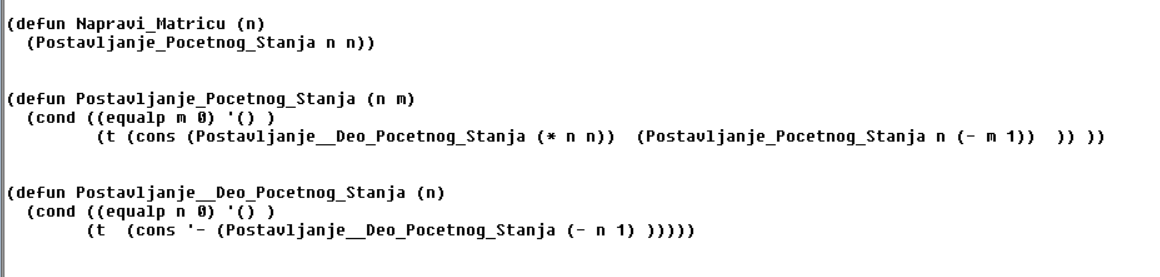
Funkcija za pokretanje igre:



Funkcija sakuplja informacije o igri i pokrece igru pozivanjem funkcije **igra.**

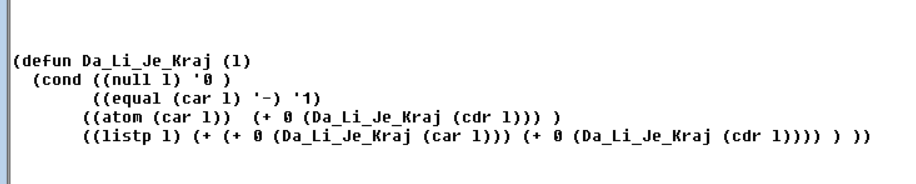
Funkcija za postavljanje početnog stanja na osnovu zadate veličine kocke:

Sastoji se od 3 funkcije,od kojih funkcija **Postavljanje\_Pocetnog\_Stanja** je zaduzena za postavljanje dubine (podlisti u listu), tj koliko puta se funkcija **Postavljanje\_Deo\_Pocetnog\_Stanja** poziva, dok je funkcija **Postavljanje\_Deo\_Pocetnog\_Stanja** zaduzena da napravi n\*n listu.



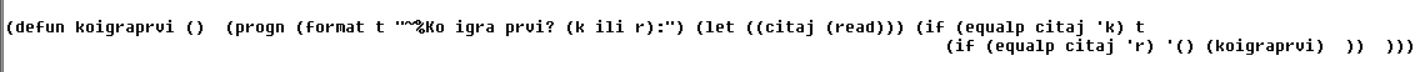
Funkcije za testiranje kraja igre:

Testira u svakoj podlisti da li postoji karakter “ - ” koji predstavlja prazno polje, pri prvom pojavljivanju ovog karaktera izlazi iz podliste i vraca 1, u suprotnom vraca 0 . Na kraju kada prodje sve podliste, vrsi se sumiranje sta je svaka podlista vratila i ako je suma 0 onda se smatra da je kraj igre, u suprotnom ako je vece od 0 onda se smatra da se igra nije zavrsila.



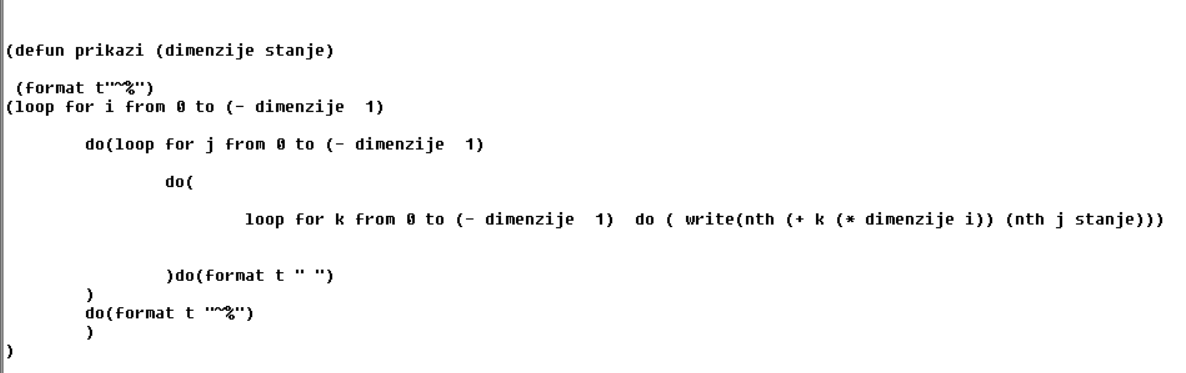
Funkcija za izbor ko će igrati prvi (čovek ili računar):

Ova funkcija stampa korisniku pitanje “Ko igra prvi” i u zavisnosti od toga sta je uneto sa tastature vraca T ili ‘(), T vraca ako je uneto k tj da korisnik igra prvi.



Funkcija koja obezbeđuje prikaz trenutnog stanja problema (igre):

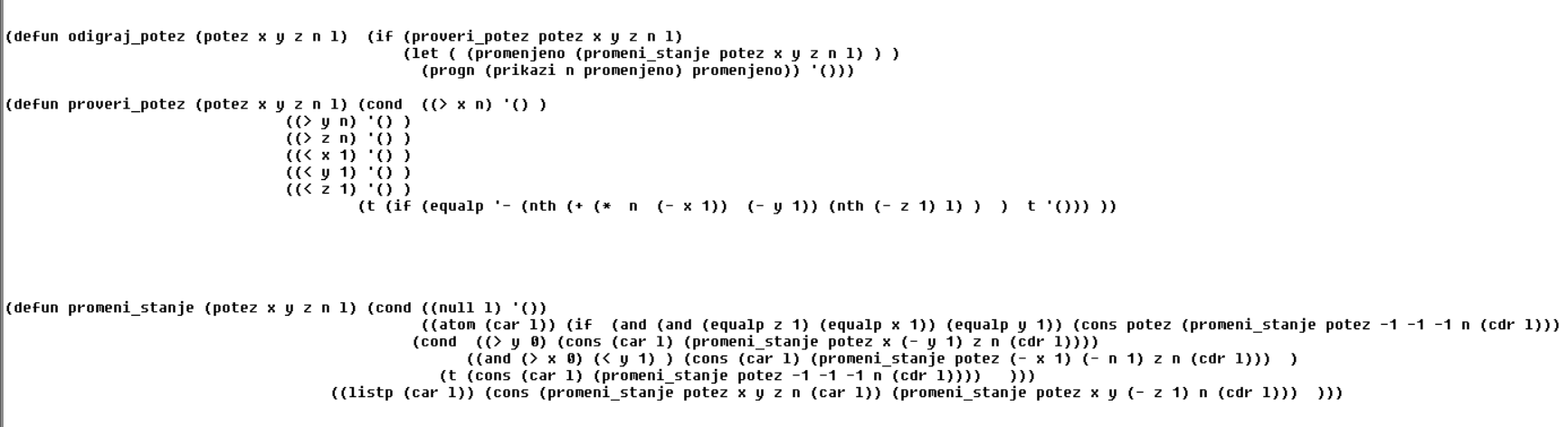
Funkcija prikazuje stanje pomocu tri for loop-a.Prvi regulise koja tabla se obradjuje, dok druga dva stampaju simbole specificnim redosledom - tako da jedna tabla izgleda na ekranu kao sto bi izgledala iz ugla jednog igraca(npr. jedna uspravna kolona je jedan stapic).Stampa se svaki k+(n\*i)-ti element aktuelne table, pri cemu n predstavlja dimenzije kocke. Ovim se postize realistican prikaz stanja igre.



Funkcije koje na osnovu zadatog poteza igrača, u obliku vrste,kolone i dubine (x y z), omogućavaju:

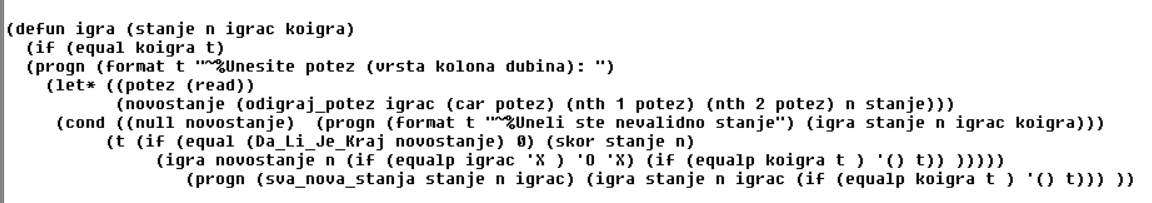
1. Proveru da li je potez valjan
2. Ako jeste, promenu prosleđenog stanja problema (igre) odigravanjem poteza.

Funkcija **odigraj\_potez** sadrzi 2 pomocne funkcije i na kraju stampa novo stanje, pomocu funkcije **proveri\_potez** ispituje da li je na unete koordinate prazno polje ili nije, i vraca T ili '(), ako jeste prazno polje onda se u promenjivoj **promenjeno** postavlja rezultat funkcije **promeni\_stanje**, koja postavlja novo stanje. Funkcija**promeni \_stanje**menja stanje samo ukoliko su sve koordinate jednake jedinici (1), Z koordinata se smanjuje prilikom prelaska na novu podlistu, X i Y koordinate se smanjuju na taj nacin kako bi zadovoljili nase predstavljanje stanja, gde je prva kolona: n\*(0 do n-1) +1, druga kolona: n\*(0 do n-1)+2 itd. Postavljanje X,Y,Z u pozivima na -1 sluzi kako bi se izbegle dalje uslovne promenjive posle postavljanja nove vrednosti na ciljano polje, i nastavilo sa pribavljanjem predhodnih vrednosti ostalih polja.



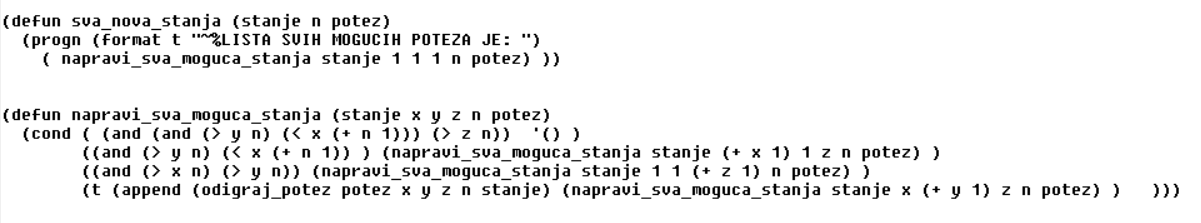
Funkcije koje obezbeđuju odigravanje partije između dva igrača (dva čoveka, ne računara i čoveka) :

Ova funkcija proverava ko igra (racunar ili korisnik) i ako je računar onda se ZA SADA poziva funkcija **sva\_nova\_stanja** koja stampa sva moguca nova stanja, i poziva se opet funkcija **igra** sa promenom ko igra, i znaka igre (X, O). Ako je korisnik cita se potez koji se unosi, onda se poziva funkcija **odigraj\_potez** gde se odigrava potez ako je validan i vraća novo stanje, onda sledi provera da li je kraj igre, tj poziva se funkcija **Da\_Li\_Je\_Kraj**, i ako jeste poziva se funkcija **skor** koja izračunava ukupan skor igre.

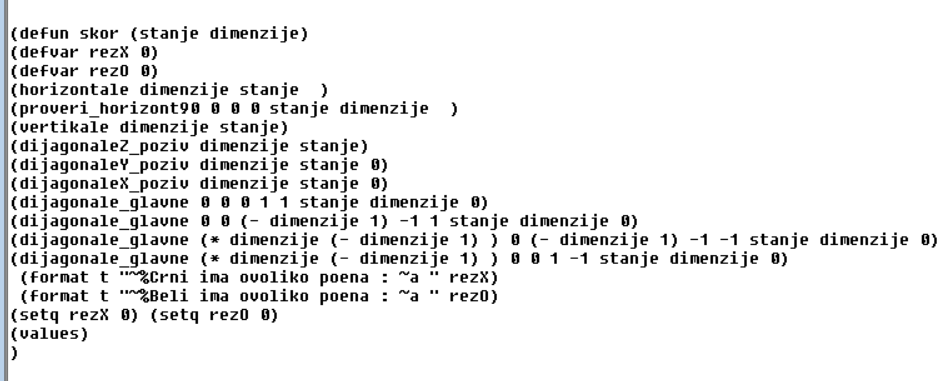


Funkcija za pravljenje liste svih mogućih poteza:

Funkcija **sva\_nova\_stanja** sluzi kao interfejs za funkciju **napravi\_sva\_moguca\_stanja** koja dok nisu (x,y,z) =0 ide redom i poziva funkciju **odigraj\_potez** i povezuje sva moguća stanja u jednu list. Trenutno samo prikazuje tj. stampa sva moguća stanja (svi potezi) na ekran.



Funkcija za odredjivanje pobednika:

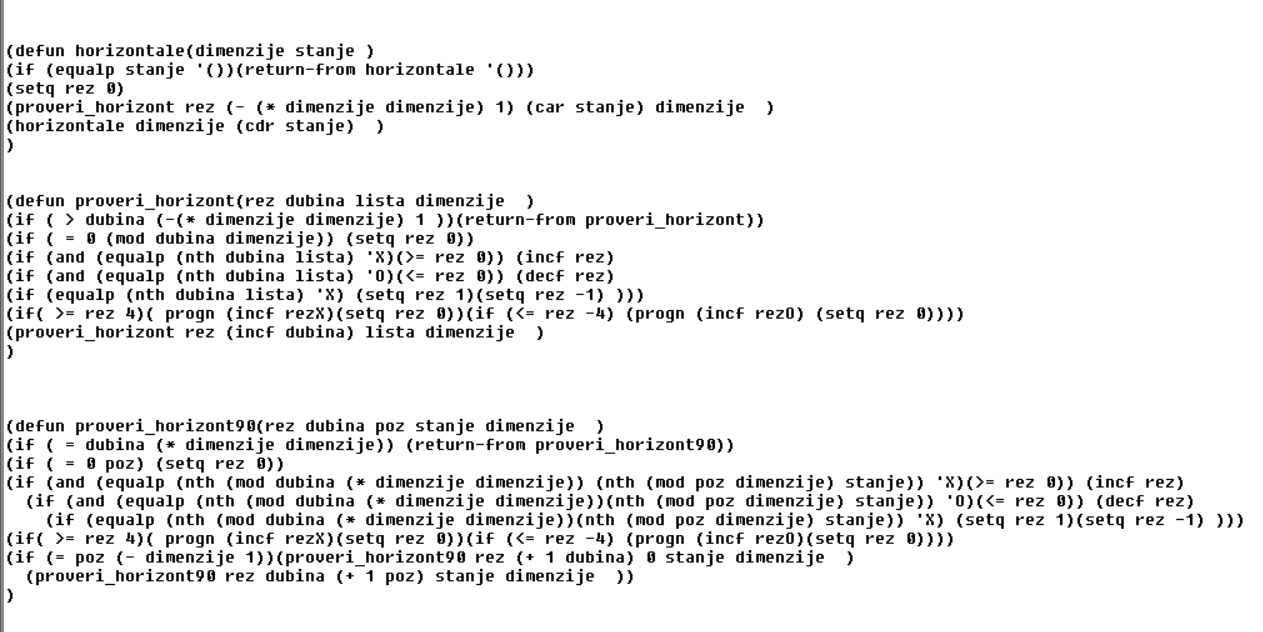


Odredjivanje pobednika se vrsi pomocu glavne funkcije i dve globalne promenljive koje predstavljaju broj postignutih poena (**rezX** i **rezO**). Glavna funkcija (**skor**) poziva pomocne, koje vrse obilazak table/kocke po vertikalama, horizontalama, i dijagonalama.  
Sve funkcije rade po slicnom principu: posmatra se niz predjenih karaktera, pri cemu se parametar **rez** povecava (ukoliko je pronadjen niz od jednog ili vise X-ova), smanjuje (O predstavljaju negativne vrednosti), ili resetuje (ako je pronadjen razlicit karakter, brojanje pocinje iz pocetka).

Vrednosti +/-4 znace da je uspesno pronadjen niz od cetiri X ili O karaktera u posmatranoj liniji, na osnovu cega se rezX i rezO povecavaju.

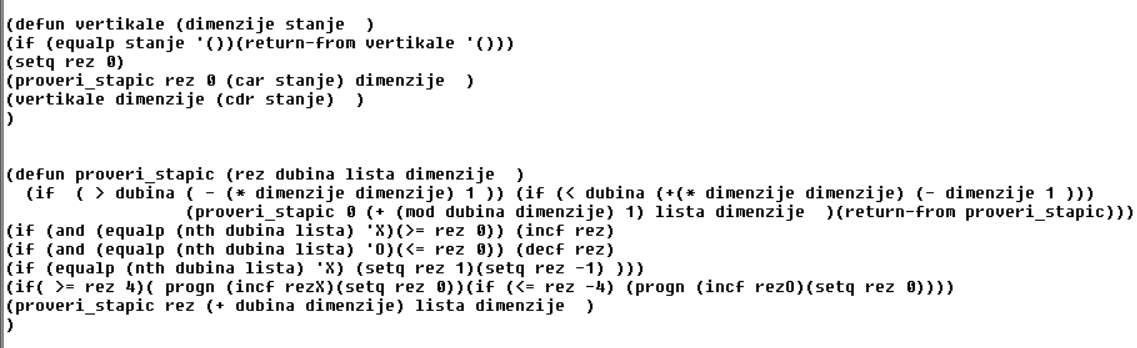
*Horizontale:*

Funkcija **horizontale**prolazi kroz glavnu listu koja predstavlja stanje igre (prenesenu kroz parametar **stanje**), za svaku podlistu poziva pomocnu funkciju (**proveri\_horizont**) koja racuna broj postignutih poena u svakoj 2D tabli predstavljenoj jednom pod-listom.  
 Druga funkcija (**proveri\_horizont90**) proverava horizontale koje idu duž table (koje bi na realnoj tabli isle od jednog igraca prema drugom), pod uglom od 90 stepeni u odnosu na prethodne.



*Vertikale:*

Vertikale se obilaze pomoću dve funkcije, **vertikale** i **proveri\_stapic.  
Vertikale** za svaku podlistu iz **stanja** pozivaju **proveri\_stapic,** koji obilaze vertikale u svakoj tabli.



*Dijagonale:*

Sporedne i glavne dijagonale kocke se obilaze na slican nacin. Za obilazak sporednih koristimo **dijagonaleX\_poziv** (takodje Y i Z), pri cemu slova X, Y, i Zoznacavaju osu koja ’probija’ ravan koju formiraju sporedne dijagonale kocke (odnosno dijagonale jedne njene stranice).   
’**\_poziv’** funkcije, kao sto im ime implicira, pozivaju vise podfunkcija (**dijagonaleX**/**Y**/**Z**) sa razlicitim parametrima. Uzet je u obzir i slucaj kada su dimenzije vece od **4**, gde dijagonalne kombinacije od 4 elementa koje nose poen mogu da se nadju van same dijagonale, paralelno njoj, pa se nakon nekoliko IF struktura funkcije pozivaju ponovo, sa adekvadnim parametrima

Glavne dijagonale kocke se obilaze zasebnom funkcijom koja se poziva cetiri puta, sa razlicitim parametrima (asc i smer, u kom smeru i da li je smer obilaska opadajuc; dubina i poz – dubina prvog elementa i podlista u kojoj se nalazi, respektivno).

