

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**VARAŽDIN**

**Jelena Valčić**

**PROBLEM STABILNIH BRAKOVA**  
**PROJEKT IZ KOLEGIJA LOGIČKO PROGRAMIRANJE**

**Varaždin, 2014.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**VARAŽDIN**

**Jelena Valčić**

**Redovni student**

**Broj indeksa: 41648/12-R**

**Smjer: Informacijsko i programsko inženjerstvo**

**Diplomski studij**

**PROBLEM STABILNIH BRAKOVA**  
**PROJEKT IZ KOLEGIJA LOGIČKO PROGRAMIRANJE**

**Mentor:**

**dr. sc. Sandra Lovrenčić, docent**

**Varaždin, 2014.**

## Sadržaj

1. Uvod .....	3
1.1. Gale-Shapely algoritam.....	3
2. Opis mogućnosti prologa.....	6
2.1. Ograničenja programa .....	16
3. Popis predikata i činjenica .....	17
4. Programski kod i komentari .....	21
5. Prikaz primjera izvođenja programa.....	30
6. Zaključak .....	37
7. Literatura .....	38



# 1. Uvod

Problem stabilnih brakova koji se pored informatičkog svijeta nalazi i u svijetu matematike i ekonomije predstavlja problem pronalaska stabilnog uparivanja dva seta elemenata prema nekom skupu preferenci za svaki element. Dakle konkretno radi se o uparivanju elemenata tako da si najbolje odgovaraju prema nekom skupu ograničenja koja dolaze sa svakim elementom, te da u konačnici ne ostane niti jedan element koji nije u paru s nekim drugim elementom.

Dakle uparivanje se odvija na način da se spajaju elementi jednog skupa sa elementima drugog skupa. Da bi uparivanje nazvali stabilnim potrebno je potvrditi kako slijedeća dva uvjeta nisu zadovoljena:

- a) Neki element A iz prvog uparenog skupa preferira neki element B iz drugog skupa uparivanja nad elementom s kojim je A već uparen
- b) B također preferira A nad elementom s kojim je B već uparen

Drugim riječima uparivanje je stabilno kada ne postoji niti jedna alternativa nekom paru (A,B) gdje bi oboje A i B boljeg para našli u nekom drugom elementu umjesto onog s kojim su već upareni.

Uobičajeno se problem stabilnih brakova definira kao :

Potrebno je vjenčati  $N$  muškaraca i  $N$  žena, gdje je pritom svaka osoba rangirala sve članove suprotnog spola sa jedinstvenim brojem između 1 i  $n$  prema redoslijedu preferiranja, na način da ne postoji dvoje ljudi suprotnog spola koji bi rađe htjeli biti u vezi nego sa partnerima s kojima već jesu u vezi. Ukoliko takve dvije osobe ne postoje brakovi su stabilni.

Algoritmi za pronalazak rješenja na ovaj problem se koriste u različitim situacijama vezanih za stvarni život ali vjerojatno najpoznatija primjena je nad studentima medicine i dodjeljivanju bolnica u kojima će raditi prema njihovim preferencama.

## 1.1. Gale-Shapely algoritam

Algoritam ima cilj pokazati kako je za bilo koji jednak broj muškaraca i žena uvijek moguće riješiti problem stabilnih brakova, te postići da su ujedno brakovi stabilni.

Algoritam uključuje nekoliko iteracija, a odvija se na slijedeći način: U prvoj iteraciji prvo svaki nezaručeni muškarac prosi ženu koju najviše preferira, te zatim svaka žena odgovara sa „Možda“ svom udvaraču kojeg najviše preferira, a svim ostalim udvaračima odgovara sa „Ne“. Ona je zatim privremeno zaručena za udvarača kojeg do sada najviše preferira, te je također taj udvarač jednako privremeno zaručen za nju. U svakoj slijedećoj iteraciji prvo svaki nezaručeni muškarac prosi ženu koju najviše preferira a do sada ju nije zaprosio (bez obzira na to da li je već zaručena). Zatim svaka žena odgovara sa „Možda“ svom udvaraču kojeg najviše preferira (bio on trenutni zaručnik ili netko drugi) a ostale odbija (to može uključivati i njenog trenutnog zaručnika). Trenutne zaruke čuvaju pravo već zaručene žene da se zamijeni i tako napusti dosadašnjeg partnera.

Što ovaj algoritam osigurava? Osigurava dvije stvari :

**Svatko je vjenčan** – to znači da kada je žena jednom zaručena ona je uvijek zaručena za nekoga. Stoga na kraju ne postoji muškarac i žena koji nisu zaručeni budući da je muškarac u nekom trenutku morao zaprositi (čak i svaku ženu ukoliko je potrebno) a kako je žena nezaručena reći će „Da“.

**Brak je stabilan** – uzmimo za primjer Anu i Peru koji su oboje zaručeni ali ne jedno za drugo. Završetkom algoritma nije moguće da i Ana i Pero preferiraju jedno drugo nad svojim trenutnim partnerima. Ukoliko Pero preferira Anu nad svojom trenutnom partnericom Anu je morao zaprositi prije svoje trenutne partnerice. Ukoliko je Ana prihvatila njegovu prosidbu a u konačnici nije udana za Peru znači da ga je ostavila zbog nekog koga više preferira, te time pokazuje kako joj se Pero ne sviđa više od njenog trenutnog partnera, a ako je odbila prosidbu znači da je već bila sa partnerom koji joj se više sviđa.

„Problem stabilnih brakova“ je implementiran u deklarativnom programskom jeziku Prolog putem Gale-Shapely algoritma, a implementacija je provedena u tri koraka kako je to specifično za deklarativne programske jezike: opis problema na način koji je svojstven jeziku, upit kojim programu govorimo što je to točno što želimo doznati i popis predikata i njihovih opisa.

Konkretno razlike u realizaciji nema u usporedbi s algoritmom pomoću kojeg se rješava problem. Što se tiče interakcije korisnik ima mogućnost učitati već postojeće datoteke muškaraca i žena s njihovim preferencama ili unijeti vlastite parove s preferencama te se ti podaci spremaju u datoteku iz koje se kasnije čita. Pored toga se pri pokretanju glavnog dijela programa korisniku

se ispisuju podaci pročitani iz datoteke, inicijalne zaruke, nova uparivanja, te u konačnici vjenčani parovi.

Način realizacije se ne razlikuje od same definicije algoritma i prati ga po koracima kojima algoritam rješava problem. Rješavanje putem računala doduše ubrzava proces uparivanja koji se svodi na sekundu ili dvije koji bi inače ručno trajao poprilično dugo. Zbog mojih ograničenih sposobnosti s Prologom sama implementacija je jednostavna bez kombiniranja sa drugim programskim jezicima. Također umjesto ručnog unošenja podataka korisnik učitava unaprijed pripremljene podatke iz datoteke doduše ukoliko želi unijeti vlastite podatke to mu je omogućeno na način da unese podatke koji se spremaju u datoteku koja se zatim isto učitava. Sučelje nudi jednostavan ispis koji prikazuje tok izvođenja programa i stanja na ključnim dijelovima algoritma. Naredbe kojima korisnik ostvaruje navedene funkcionalnosti su predikati u prologu koji se pozivaju odabirom određenih komandi. Svi predikati i njihove mogućnosti će biti detaljno objašnjeni i opisani u nastavku rada, također će biti prikazi rada programa i primjer datoteke s podacima i izvođenja programa.

## 2. Opis mogućnosti prologa

U ovom će se poglavlju opisati mogućnosti SWI Prologa i kako se one uklapaju u predikate programa kako bi ostvarili potrebne funkcionalnosti.

Za početak kako bi korisnik mogao znati koje su funkcionalnosti i kako se kretati programom potrebno je ostvariti nekakav izbornik koji se poziva na početku programa na način da se sa komandne linije pozove predikat **izbornik**.

**izbornik :-**

```
    cls,  
    writeln('PROBLEM STABILNIH BRAKOVA'),  
    writeln('~~~~~IZBORNIK~~~~~'),  
    writeln('a. Ucitaj datoteku'),  
    writeln('b. Upisi vlastite preference '),  
    writeln('c. Pomoc'),  
    writeln('d. Izlaz'),  
    write('Vas izbor: '), read(Izbor),  
    (Izbor=='a' -> (trazi_datoteku),!;  
    (Izbor=='b' -> (cls, unos),!;  
    (Izbor=='c' -> (cls, hlp),!;  
    (Izbor=='d' -> halt)))).
```

U izborniku se koristi predikat **cls** koji služi za brisanje teksta koji se nalazio na ekranu prije pokretanja programa kako bi se dobio čist ekran za rad programa. Budući da je korišten SWI Prolog ali na Windows operacijskom sustavu nije se mogao koristiti ugrađeni predikat **shell/1** napravljen je poseban predikat **cls** koji služi za istu svrhu ali sadrži posebne znakove za čišćenje ekrana na Windows operacijskom sustavu budući da terminali na Windows operacijskom sustavu ne podržavaju biblioteku koju podržava Linux operacijski sustav.



**cls :-**

**write("\e[H\e[2J").**

Kako bi se koristili ovi posebni znakovi upotrijebljen je ugrađeni predikat **write/1** koji ispisuje zadani tekst, a nalazi se u modulu **[basics]**. Također se taj predikat koristi i u samom izborniku kako bi se ispisao izbornik korisniku. Korišten je i ugrađeni predikat **writeln/1** koji ispisuje zadani tekst i prelazi u novi red, te se i on nalazi u modulu **[basics]**. Još jedan predikat iz tog modula je **read/1** koji služi za čitanje korisnikovog unosa. Također se koristi i predikat **!** odnosno rez, kojim se reže postupak traženja zadovoljavajućeg rješenja nakon što se nađe rješenje koje je traženo, odnosno prekida se traženje alternativnih rješenja nakon što se nađe odgovarajuće rješenje. U ovom slučaju osigurava da Prolog ne dođe u beskonačnu petlju. Na kraju se kod izbora za izlaz se poziva ugrađeni predikat **halt/1** kako bi se prekinuo rad programa, a dio je modula **[basics]**.

Nakon što se odabere prva funkcionalnost poziva se predikat **trazi\_datoteku** koja od korisnika traži unos naziva datoteke iz koje se čitaju podaci za rješavanje problema. Nakon što korisnik unese ime datoteke poziva se glavni predikat cijelog programa koji je zadužen za provođenje rješavanja problema stabilnih brakova imena **psb/1**.

Predikat prvo čisti ekran od viška teksta a zatim poziva predikat **ucitaj\_datoteku/1** koji koristi ugrađeni predikat **see/1** koji služi za otvaranje toka podataka odredišta kako bi se moglo iz njega čitati i uzeti ga kao trenutni unos podataka.

**ucitaj\_datoteku(D):-**

```
    see(D),
    read(Broj_parova),
    write('Ukupno muskaraca i zena: '),
    X is Broj_parova * 2,
    writeln(X),
    write('Moguće parova: '),
    writeln(Broj_parova),
    nl,
    writeln('Muskarci i preference:'),
    dohvati_podatke(Broj_parova, Broj_parova, musko),

    nl,
    writeln('Žene i preference:'),
    dohvati_podatke(Broj_parova, Broj_parova, zensko),
    nl,
    assert(parovi(Broj_parova)),
    seen.
```

Unutar predikata **ucitaj\_datoteku/1** se poziva predikat **dohvati\_podatke/3** i to u dvije verzije, jedna za muškarce druga za žene. Predikat u obje verzije izvodi iste korake samo što u jednoj iz datoteke učitava podatke za muškarce a u drugoj učitava podatke za žene.

```

dohvati_podatke(_, 0, _).
dohvati_podatke(Broj, Broj2, musko):-
    read(Ime),
    dohvati_preference(Broj,L),
    assert(preferira(Ime,L)),
    assert(musko(Ime)),
    assert(samac(Ime)),
    write('\t'), write(Ime), write(' '), writeln(L),
    Broj3 is Broj2-1,
    dohvati_podatke(Broj, Broj3, musko).

```

Unutar samog predikata se koristi ugrađeni predikat **assert/1** koji služi kako bi se pročitani podaci spremili u varijable primjerice **assert(musko(Pero))**. Ukoliko se kontrolni broj smanji na nulu i nemamo više podataka za zapisati gotov je rad predikata. Kontrolni broje unesen na početku datoteke podataka koji označava koliko parova se nalazi u datoteci kako bi se mogla odvrstiti petlja zapisivanja. Unutar samog predikata se još poziva predikat **dohvati\_preference/2** koji je zadužen da se za svaku osobu čitaju preference iz datoteke i zapisuju u odgovarajuću listu. Izvođenje predikata se prekida kada je kontrolni broj dosegao nulu. Za oba predikata kontrolni broj omogućava izvedbu for petlje i osiguravanje da petlja ne ode u beskonačnost.

```

dohvati_preference(0,[]).
dohvati_preference(Broj, [H|L]):-
    read(H),
    Broj2 is Broj-1,
    dohvati_preference(Broj2,L).

```

Nakon što navedeni predikati odrade svoju zadaću se pomoću ugrađenog predikata **assert/1** zapisuje i kontrolni broj, te se pomoću ugrađenog predikata **seen/0** zatvara tok podataka prema datoteci i prebacuje se na standardni user input tok podataka.

Predikat **psb/1** zatim slijedno poziva predikat **upari\_samce/0** čija je zadaća odraditi prvi korak algoritma koji rješava problem stabilnih brakova.

**upari\_samce:-**

```
    samac(Musko),  
    samac(Zensko),  
    musko(Musko),  
    zensko(Zensko),  
    assert(zaruceni(Musko,Zensko)),  
    write('Trenutno u vezi: '),  
    write(Musko), write(' i '), writeln(Zensko),  
    retract(samac(Musko)),  
    retract(samac(Zensko)),  
    upari_samce.
```

**upari\_samce:-**

```
    nl,  
    zaruci.
```

Predikat uzima samce iz baze činjenica, redom muškarca i ženu te ih korištenjem ugrađenog predikata **assert/1** dodaje u činjenicu **zaruceni**, time se ostvaruje prvi korak algoritma gdje svaki slobodan muškarac redom prosi ženu koju najviše preferira. Žena zatim odgovara možda, te postaje privremeno zaručena za muškarca koji ju prvi pita a ostale odbija. Pomoću ugrađenog predikata **retract/1** zaručenog muškarca i ženu mičemo iz popisa samaca. Nakon toga se poziva predikat **zaruci/0** koji izvršava drugi korak algoritma.

**zaruci:-**

```
zaruceni(Musko1, Zensko1),
zaruceni(Musko2, Zensko2),
nadj_preferencu(Musko1, Zensko2, On_preferira),
nadj_preferencu(Musko1, Zensko1, On_preferira_staro),
nadj_preferencu(Zensko2, Musko1, Ona_preferira),
nadj_preferencu(Zensko2, Musko2, Ona_preferira_staro),
On_preferira < On_preferira_staro,
Ona_preferira < Ona_preferira_staro,
retract(zaruceni(Musko1, Zensko1)),
retract(zaruceni(Musko2, Zensko2)),
write('Novi par: '),
write(Musko1), write(' i '), writeln(Zensko2),
write('Novi par: '),
write(Musko2), write(' i '), writeln(Zensko1),
nl,
assert(zaruceni(Musko1, Zensko2)),
assert(zaruceni(Musko2, Zensko1)),
zaruci.
```

**zaruci:-**

```
not(samac(_)).
```

Uzimaju se dva para iz činjenice **zaruceni**, te kako bi se pokrili svi slučajevi gledaju se preference muškarca prvog para vezi prema ženi drugog para i prema ženi za koju je trenutno zaručen. Također se gledaju preference žene drugog para prema muškarcu prvog para i muškarcu za kojeg je trenutno zaručena. Kako bi odredili koju ženu muškarac više preferira i obrnuto koristi se predikat **nadj\_preferencu/3** koji primjerice za muškarca traži u njegovoj listi preferenci na kojem se mjestu nalazi žena za koju je zaručen ili žena iz drugog para. U slučaju

ovog programa težine preferenci se gledaju s lijeva na desno gdje je prva osoba s lijeva najpoželjnija, a osoba prva s desna najmanje preferirana.

**nadji(Osoba, [Osoba|\_], 1).**

**nadji(Osoba, [\_|L], Stupanj\_preferiranja):-**

**nadji(Osoba, L, Broj),  
Stupanj\_preferiranja is Broj+1.**

**nadji\_preferencu(lme1, lme2, Preferenca):-**

**preferira(lme1, Lista\_preferenci),  
nadji(lme2, Lista\_preferenci, Preferenca).**

Predikat **nadji\_preferencu/3** radi na način da iz baze činjenica dohvaća preference za muškarca u pitanju i poziva predikat **nadji/3** koji za ženu, za koju se traži preferenca, gleda u listi preferenci muškarca na kojoj se ona poziciji nalazi i vraća stupanj preferiranja. Stupanj preferiranja se bilježi na način da ukoliko se žena ne nalazi na prvoj poziciji s lijeva onda se početni broj koji je jedan povećava za jedan sve dok se ne dođe do pozicije u listi gdje se nalazi tražena žena. Kada je žena nađena prekida se izvođenje predikata i nastavlja se izvođenje predikata **zaruci/0**. Nakon što se provjere preference za oba slučaja one se uspoređuju na način da se gleda da li je stupanj preferiranja za ženu iz drugog para manji od stupnja preferiranja za ženu za koju je muškarac zaručen, te isto tako i za muškarca. Razlog takve usporedbe je taj da je stupanj preferiranja manji što je osoba poželjnija, kako je to već objašnjeno. Ukoliko je stupanj preferiranja za ženu iz drugog para manji od onog za trenutno zaručenu ženu i stupanj preferiranja muškarca iz prvog para manji od stupnja preferiranja muškarca za kojeg je žena drugog para zaručena, zaruke se raskidaju i muškarac prvog para i žena drugog para se zaručuju. Ekvivalentno tome žena prvog para i muškarac drugog para se također zaručuju.

Kako bi se pojasnio proces možemo reći ovako: ukoliko muškarac kojeg se promatra više preferira ženu zaručenog za nekog drugog muškarca u usporedbi sa ženom za koju je trenutno zaručen, te istovremeno ta žena više preferira muškarca kojeg se promatra nego muškarca za

kojeg je zaručena oni raskidaju zaruke i ostavljaju svoje partnere kako bi se zaručili. Stari parovi se brišu iz činjenice **zaruceni** te se dodaju novi. Bitno je napomenuti kako će u prvom prolazu oba para biti ista pa se tu ništa značajno neće dogoditi. Jedina razlika u usporedbi sa algoritmom jest da se proces zaručivanja odvija na način da ispitujemo da li pojedinci između dva para preferiraju jedno drugo a ne svoje zaručnike umjesto da u drugoj iteraciji svaki slobodan muškarac prosi ženu koju najviše preferira bez obzira na zaruke, te tako uparuje pojedince. Predikat **zaruci/0** se ponavlja dok se ne isprobaju sve moguće kombinacije kako bi se osiguralo stabilno uparivanje, odnosno kako bi se osiguralo da ne postoji muškarac i žena koji bi rađe bili jedan s drugim nego sa svojim partnerom.

Nakon što se nađu odgovarajući parovi unutar predikata **psb/1** poziva se predikat **vjencaj/0** koji u osnovi samo potvrđuje konačne parove. Konkretno miče parove iz činjenice **zaruceni** i sprema ih u činjenicu **vjencani** te ih ispisuje korisniku kao informaciju o konačnim parovima. Cijeli proces se izvodi sve dok ne postoji više zaručenih parova. Za provjeru postojanja zaručenih parova ili samaca se koristi ugrađeni predikat **not/1** koji provjerava postojanje.

**vjencaj:-**

```
zaruceni(Musko, Zensko),  
retract(zaruceni(Musko, Zensko)),  
assert(vjencani(Musko, Zensko)),  
write('\t'), write(Musko), write(' i '), write(Zensko), nl,  
vjencaj.
```

**vjencaj:-**

```
not(zaruceni(_,_)).
```

Pored ovih predikata koji ostvaruju glavnu funkcionalnost programa postoji mogućnost unosa vlastitih podataka koji se spremaju u datoteku koja se kasnije prilikom korištenja glavne funkcionalnosti može učitati. Iz predikata **izbornik/0** se odabirom druge po redu funkcionalnosti zove predikat **unos/0** koji omogućuje unos vlastitih preferenci.

**unos:-**

```
write('Unesite naziv datoteke: '), read(Datoteka),  
nl,  
write('Unijeti broj parova: '), read(Broj_parova),  
assert(parovi(Broj_parova)),  
nl,  
string_concat(Datoteka, '.txt', Datoteka2),  
open(Datoteka2, append, S),  
write(S,Broj_parova),  
write(S,'\n'),  
writeln('Unesite muskarce i njihove preference: '),  
zapisi_podatke(Broj_parova, S),  
nl,  
writeln('Unesite zene i njihove preference: '),  
zapisi_podatke(Broj_parova, S),  
close(S),  
nl,  
write('Vasi podaci su zapisani u datoteku '),  
writeln(Datoteka2),  
nastavak_programa.
```

U ovom se predikatu koriste pored već objašnjenih ugrađenih predikata koriste i **string\_concat/3** koji omogućuje nevezanom argumentu spajanje sa string objektom u ovom slučaju konkretno omogućuje povezivanje imena datoteke koju korisnik unese sa njegovom ekstenzijom u novu varijablu kako bi dobili cijeli naziv datoteke. Slijedeći je **open/3** koji otvara tok podataka prema unesenom odredištu (datoteka ili cjevovod) u kojeg zapisujemo podatke koje korisnik unosi uz određeni način rada u ovom slučaju je to *append* koji omogućuje dodavanje zapisa na kraj datoteke. Također se koristi predikat **close/1** koji zatvara tok podataka koji smo specificirali.



Pored navedenih predikata koristi se predikat **zapisi\_podatke/2** pomoću kojeg se upisuju podaci za muškarce, a zatim žene.

```
zapisi_podatke(0,_).
```

```
zapisi_podatke(Broj_parova, S):-
```

```
    write('Osoba i preference: '),  
    read_line_to_codes(user_input, Input),  
    string_to_atom(Input,IA),  
    write(S,IA),  
    write(S,'\n'),  
    Broj_parova2 is Broj_parova-1,  
    zapisi_podatke(Broj_parova2, S).
```

U predikat ulaze podaci o broju parova kako bi se mogla ostvariti svojevrsna for petlja kao što je to slučaj i kod predikata **dohvati\_podatke/3** kako bismo unijeli odgovarajući broj elemenata i ujedno kontrolirali petlju da ne ode u beskonačnost budući da je postavljen uvjet da se prekida izvođenje kada broj parova dođe do 0. U ovom se predikatu koriste još dva ugrađena predikata od kojih je jedan **read\_line\_to\_codes/2** koji služi za čitanje linije korisničkog unosa u tok podataka te ga spaja u varijablu gdje se spremaju podaci i završava liniju sa oznakom za novi red. Drugi je **string\_to\_atom/2** koji služi za konverziju teksta u atome. Na završetku predikata **unos/0** se poziva predikat **nastavak\_programa/0** koji služi kao pauza da bi vidjeli ispise tokom programa.

Posljednja funkcionalnost je poziv predikata **hlp/0** koji služi za ispis bitnih informacija oko ograničenja, korištenja i specifičnosti programa. Ukoliko korisnik želi izaći iz programa u izborniku treba odabrati funkcionalnost koja poziva ugrađeni predikat **halt/0** koji služi za terminiranje izvršavanja prologa.

Za sve rekurzije su definirane granične vrijednosti koje će spriječiti da procedura pozove samu sebe ukoliko je ispunjen definirani kriterij. Također je svakim pozivom same sebe osigurano da se procedura približi temeljnom kriteriju. Način na koji je to ostvareno i iz kojeg razloga je to tako je već obrazloženo tokom opisa mogućnosti Prologa.

## 2.1. Ograničenja programa

Postoje poneka ograničenja za rad programa, te poseban način unošenja podataka koji će biti u nastavku navedeni kako bi se lakše znalo koristiti programom. Isti podaci se nalaze u funkcionalnosti Pomoć koju korisnik može odabrati unutar programa.

1. Odnos preferenci:

Najpoželjniji/a -----> Najnepoželjniji/a

Lijevo -----> Desno

2. Učitavanje datoteke

Potrebno je upisati naziv zajedno sa ekstenzijom unutar jednostrukih navodnika.

npr. 'test.txt'

3. Unos osobe i njenih preferenci:

Prvo se unosi osoba za koju želimo dodati preference, a zatim slijede preference redom od najpoželjnijeg partnera do najnepoželjnijeg partnera. Pritom je potrebno nakon svakog imena staviti točku.

npr. pero. ana. sanja. iva.

4. Skup podataka

Da bi program radio kako treba potrebno je imati jednak broj muškaraca i žena, kao što je i opisano kod algoritma.

### 3. Popis predikata i činjenica

Predikat	Opis
<b>cls</b>	Predikat koji služi za „čišćenje“ ekrana.
<b>izbornik</b>	Predikat koji služi za ispis izbornika s četiri mogućnosti: Učitaj datoteku, Upiši vlastite preference, Pomoć i Izlaz iz programa.
<b>trazi_datoteku</b>	Predikat koji traži unos imena datoteke s podacima i poziva predikat <b>psb/1</b> .
<b>nastavak_programa</b>	Predikat koji služi za pauziranje programa kako bi se vidio ispis podataka na ekranu, te traži unos znaka za nastavak programa.
<b>psb(D)</b>	Predikat koji za uneseno ime datoteke ( <b>D</b> ) s podacima izvršava korake rješavanja problema stabilnih brakova.
<b>ucitaj_datoteku(D)</b>	Predikat koji za ime datoteke( <b>D</b> ) učitava podatke iz nje.
<b>dohvati_preference(Broj, L)</b>	Predikat koji kao podatak dobiva broj parova ( <b>Broj</b> ) kao kontrolu petlje, te za pojedinu osobu čita preference iz datoteke podataka i zapisuje ih u listu ( <b>L</b> ).
<b>dohvati_podatke(Broj, Broj2, zensko)</b>	Predikat koji kao ulazne podatke dobiva broj parova ( <b>Broj</b> ) koji se prosljeđuje predikatu <b>dohvati_preference/2</b> za kontrolu petlje i ( <b>Broj2</b> ) koji služi za kontrolu petlje ovog predikata kako bi se pročitale sve osobe iz datoteke s podacima. Taj se broj smanjuje nakon svakog pročitanoj reda dok ne dođe do 0, te se tada prekida čitanje podataka. Zadnji parametar je ime činjenice <b>zensko(Ime)</b> u koju ćemo

	<p>zapisati imena žena koje se nalaze u datoteci s podacima i definirati da su ženske osobe. Tokom izvođenja se zapisuje u činjenicu <b>preferira(Ime,L)</b> zapisuje osoba i njene preference, te se u činjenicu <b>samac(Ime)</b> zapisuje osoba kao samac.</p>
<b>dohvati_podatke(Broj, Broj2, musko)</b>	<p>Predikat koji kao ulazne podatke dobiva broj parova (<b>Broj</b>) koji se prosljeđuje predikatu <b>dohvati_preference/2</b> za kontrolu petlje i (<b>Broj2</b>) koji služi za kontrolu petlje ovog predikata kako bi se pročitale sve osobe iz datoteke s podacima. Taj se broj smanjuje nakon svakog pročitanoog reda dok ne dođe do 0, te se tada prekida čitanje podataka. Zadnji parametar je ime činjenice <b>musko(Ime)</b> u koju ćemo zapisati imena muškaraca koje se nalaze u datoteci s podacima i definirati da su muške osobe. Tokom izvođenja se zapisuje u činjenicu <b>preferira(Ime,L)</b> zapisuje osoba i njene preference, te se u činjenicu <b>samac(Ime)</b> zapisuje osoba kao samac.</p>
<b>upari_samce</b>	<p>Predikat provodi prvi korak uparivanja muškaraca i žena. Pritom se osobe brišu iz činjenica <b>samac(Ime)</b> a dodaju kao parovi u činjenicu <b>zaruceni(Musko,Zensko)</b>. Nakon uparivanja zove se predikat <b>zaruci/0</b>.</p>
<b>nadji(Osoba, L, Stupanj_preferiranja)</b>	<p>Predikat koji na za parametre dobiva ime osobe za koju se traži stupanj preferiranja i listu preferenci na način da se traži pozicija u listi koja se bilježi u varijablu <b>Broj</b> koji se svakim pomakom u desno povećava za jedan budući da je određeno da će najpoželjnija osoba biti prva s lijeva i time imati stupanj preferiranja 1. Prekida se izvođenje kada nađemo poziciju te se <b>Broj</b> vraća kao stupanj preferiranja.</p>
<b>nadji_preferencu(Ime1, Ime2, Preferenca)</b>	<p>Predikat koji za muškarca/ženu (<b>Ime1</b>) uzima listu preferenci iz činjenice <b>preferira(Ime, L)</b> i poziva predikat <b>nadji(Osoba, L, Stupanj_preferiranja)</b></p>

	pomoću kojeg će vratiti stupanj preferiranja.
<b>zaruci</b>	<p>Predikat koji provodi drugi korak rješavanja problema stabilnih brakova. Uzima dva para i pomoću predikata <b>nadji_preferencu(Ime1, Ime2, Preferenca)</b> traži stupanj preference za osobe u parovima. Uspoređuju se stupnjevi preferenci muškarca prvog para i žene drugog para na način da se gleda da li je stupanj preferiranja muškarca/žene za koje se gleda veći od stupnja preferenci za njihove trenutne partnere. Odnosno jednostavnije, da li muškarac prvog para i žena drugog para međusobno više preferiraju jedno drugo u usporedbi s njihovim trenutnim partnerima. Ukoliko je to slučaj zapisuju se novi parovi u činjenicu <b>zaruceni(Musko, Zensko)</b>, a stari se brišu. Predikat se izvršava sve dok više nema samaca i ne postoje muškarac i žena koji se međusobno više preferiraju u usporedbi s njihovim partnerima.</p>
<b>vjencaj</b>	<p>Predikat koji služi kao potvrda konačnih uparivanja i briše parove iz činjenice <b>zaruceni(Musko, Zensko)</b> te ih zapisuje u činjenicu <b>vjencani(Musko, Zensko)</b>. Parove ispisuje na ekran kako bi korisnik vidio konačne parove. Izvođenje se ponavlja sve dok ima elemenata i činjenici <b>zaruceni(Musko, Zensko)</b>.</p>
<b>unos</b>	<p>Predikat koji od korisnika zahtjeva unos naziva datoteke u koju će se spremati podaci o parovima i preferencama. Također od korisnika zahtjeva unos broja parova kako bi se mogla ostvariti kontrola petlje da predikat ne bi otišao u beskonačnu petlju. Nakon toga se poziva predikat <b>zapisi_podatke(Broj_parova, S)</b> koji će zapisati osobu i njezine preference u datoteku.</p>
<b>zapisi_podatke(Broj_parova, S)</b>	<p>Predikat koji kao parametar dobiva broj parova kako bi se kontrolirala petlja i tok podataka. Traži korisnika za upiše osobu i njegove preference, te se red zajedno sa prijelazom u novu liniju zapisuje u datoteku.</p>

<b>hlp</b>	Predikat koji ispisuje bitne podatke oko izvođenja programa, njegova ograničenja i način unosa određenih podataka.
------------	--

Tablica 3.1. Popis predikata s argumentima i njihovim opisom

Činjenica	Opis
<b>parovi(Broj_parova)</b>	<b>Broj_parova</b> označava broj parova koji se nalaze u datoteci s podacima
<b>preferira(Ime,L)</b>	<b>Ime</b> označava ime osobe, <b>L</b> je lista preferenci vezana za osobu
<b>zensko(Ime)</b>	<b>Ime</b> označava ime ženske osobe
<b>musko(Ime)</b>	<b>Ime</b> označava ime muške osobe
<b>samac(Ime)</b>	<b>Ime</b> označava ime osobe koja je samac
<b>zaruceni(Musko,Zensko)</b>	<b>Musko</b> označava ime muškarca koji je u vezi sa <b>Zensko</b> koje označava ime žene koja je u vezi sa <b>Musko</b>
<b>vjencani(Musko, Zensko)</b>	<b>Musko</b> označava ime muškarca koji je vjenčan sa <b>Zensko</b> koje označava ime žene koja je vjenčana sa <b>Musko</b>

Tablica 3.2. Popis činjenica s argumentima i njihovim opisom

## 4. Programski kod i komentari

**%Predikat koji služi za čišćenje ispisa s ekrana**

cls :-

```
write('\e[H\e[2J').
```

**%Predikat u kojem je definiran izbornik, te koji se predikat poziva**

**%Ovisno o odabiru kojeg korisnik napravi**

izbornik :-

```
cls,
```

```
writeln('PROBLEM STABILNIH BRAKOVA'),
```

```
writeln('~~~~~IZBORNIK~~~~~'),
```

```
writeln('a. Učitaj datoteku'),
```

```
writeln('b. Upisi vlastite preference '),
```

```
writeln('c. Pomoc'),
```

```
writeln('d. Izlaz'),
```

```
write('Vas izbor: '), read(Izbor),%cita se korisnikov izbor
```

```
(Izbor=='a' -> (trazi_datoteku),!; %poziva se predikat koji čita podatke iz odabrane
```

**datoteke**

```
(Izbor=='b' -> (cls, unos),!; %poziva se predikat kojim korisnik može zapisati
```

**vlastite podatke u datoteku koju kasnije može učitati**

```
(Izbor=='c' -> (cls, hlp),!;%poziva se predikat koji prikazuje upute za korištenje
```

**programa**

```
(Izbor=='d' -> halt))). %prekida se izvođenje programa
```

**%Predikat kojim se cita naziv datoteke iz koje uziamamo parametre i**

**%Prosljedjuje se predikatu koji je glavni dio programa**

trazi\_datoteku:-

```
    nl,  
    write('Upisite naziv datoteke: '),  
    read(D),  
    psb(D).%poziva se predikat koji izvodi rjesavanje problema stabilnih brakova
```

**%Predikat kojim se obavlja jednostavan nastavak rada programa**

nastavak\_programa:-

```
    nl,  
    write('Unesite n. za nastavak... '),  
    read(X),  
    (X=='n' -> izbornik).%poziva se predikat izbornik
```

**%Predikat koji poziva kljucne predikate za rjesavanje**

**%Problema stabilnih brakova**

psb(D):-

```
    cls,  
    učitaj_datoteku(D),%predikat koji učitava podatke iz odabrane datoteke  
    upari_samce,%predikat koji obavlja uparivanje muskaraca i zena  
    write('Vrijeme za svadbu, sve najbolje mladencima!'),  
    nl,  
    vjencaj,%predikat koji potvrđuje i "vjencava" uparene muskarce i zene  
    nastavak_programa.%predikat koji izvodi nastavak programa
```



**%Predikat koji ucitava podatke iz odabrane datoteke**

ucitaj\_datoteku(D):-

see(D),

read(Broj\_parova),%cita se broj parova kako bi se znalo koliko je muskaraca a koliko

**zena**

write('Ukupno muskaraca i zena: '),

X is Broj\_parova \* 2,

writeln(X),

write('Moguće parova: '),

writeln(Broj\_parova),

nl,

writeln('Muskarci i preference:'),

dohvati\_podatke(Broj\_parova, Broj\_parova, musko),%poziva se predikat koji cita i

**zapisuje muskarce i njihove preference**

nl,

writeln('Zene i preference:'),

dohvati\_podatke(Broj\_parova, Broj\_parova, zensko),%poziva se predikat koji cita i

**zapisuje muskarce i njihove preference**

nl,

assert(parovi(Broj\_parova),%zapisuje se koliko je parova

seen.

dohvati\_preference(0,[]).

**%Predikat koji dohvaca preference i sprema ih u listu**

dohvati\_preference(Broj, [H|L]):-

read(H),

Broj2 is Broj-1,%broj se smanjuje kako bi procitali sve preference

dohvati\_preference(Broj2,L).

```
dohvati_podatke(_, 0, _).
```

#### **%Predikat koji dohvaca zene i njihove preference**

```
dohvati_podatke(Broj, Broj2, zensko):-
```

```
    read(Ime),%cita se ime osobe cije preference trazimo
```

```
    dohvati_preference(Broj,L),%citamo preference trenutne osobe
```

```
    assert(preferira(Ime,L)),%zapisujemo osobu i njene preference
```

```
    assert(zensko(Ime)),%zapisujemo ime zenske osobe
```

```
    assert(samac(Ime)),%zapisujemo da je osoba samac
```

```
    write('\t'), write(Ime), write(' '), writeln(L),
```

```
    Broj3 is Broj2-1,%broj se smanjuje kako bi procitali sve zene i njihove preference
```

```
    dohvati_podatke(Broj, Broj3, zensko).
```

#### **%Predikat koji dohvaca muskarce i njihove preference**

```
dohvati_podatke(Broj, Broj2, musko):-
```

```
    read(Ime),%cita se ime osobe cije preference trazimo
```

```
    dohvati_preference(Broj,L),%citamo preference trenutne osobe
```

```
    assert(preferira(Ime,L)),%zapisujemo osobu i njene preference
```

```
    assert(musko(Ime)),%zapisujemo ime muske osobe
```

```
    assert(samac(Ime)),%zapisujemo da je osoba samac
```

```
    write('\t'), write(Ime), write(' '), writeln(L),
```

```
    Broj3 is Broj2-1,%broj se smanjuje kako bi procitali sve muskarce i njihove  
preference
```

```
    dohvati_podatke(Broj, Broj3, musko).
```

**%Predikat koji uparuje samce(pocetno uparivanje)**

upari\_samce:-

```
samac(Musko),
samac(Zensko),
musko(Musko),
zensko(Zensko),
assert(zaruceni(Musko,Zensko)),%zapisujemo parove
write('Trenutno u vezi: '),
write(Musko), write(' i '), writeln(Zensko),
retract(samac(Musko)),%brisemo muskarca iz popisa samaca
retract(samac(Zensko)),%brisemo zenu iz popisa samaca
upari_samce.
```

upari\_samce:-

```
nl,
zaruci.%nakon sto se obavi pocetno uparivanje krece proces zaruka
```

nadji(Osoba, [Osoba|\_], 1).

**%Predikat koji trazi preferencu osobe i zapisuje kolika je preferenca**

nadji(Osoba, [\_|L], Stupanj\_preferiranja):-

```
nadji(Osoba, L, Broj),
Stupanj_preferiranja is Broj+1.%zapravo redni broj na listi koji se koristi za
usporedbu
```

**%Predikat koji trazi da li osoba preferira nekog drugog**

nadji\_preferencu(Ime1, Ime2, Preferenca):-

```
preferira(Ime1, Lista_preferenci),
nadji(Ime2, Lista_preferenci, Preferenca).%predikat koji trazi preferencu
```

**%Predikat koji odraduje preostala uparivanja i zarucuje muskarce i zene**

zaruci:-

zaruceni(Musko1, Zensko1),%uzima se prvi par

zaruceni(Musko2, Zensko2),%uzima se drugi par

nadji\_preferencu(Musko1, Zensko2, On\_preferira),%poziva se predikat koji trazi da  
**li muskarac preferira neku drugu zenu i koliko**

nadji\_preferencu(Musko1, Zensko1, On\_preferira\_staro),%provjerava se koliko  
**muskarac preferira zenu s kojom je trenutno**

nadji\_preferencu(Zensko2, Musko1, Ona\_preferira),%provjerava se da li zena u  
**drugom paru preferira nekog drugog muskarca i koliko**

nadji\_preferencu(Zensko2, Musko2, Ona\_preferira\_staro),%provjerava se koliko  
**zena preferira muskarca s kojim je trenutno**

On\_preferira < On\_preferira\_staro,%usporedjuju se preference muskarca iz prvog  
**para**

Ona\_preferira < Ona\_preferira\_staro,%usporedjuju se preference zene iz drugog  
**para**

retract(zaruceni(Musko1, Zensko1)),%ukoliko muskarac preferira neku drugu zenu  
**raskidaju se zaruke**

retract(zaruceni(Musko2, Zensko2)),%ukoliko zena preferira nekog drugog  
**muskarca raskidaju se zaruke**

write('Novi par: '),

write(Musko1), write(' i '), writeln(Zensko2),

write('Novi par: '),

write(Musko2), write(' i '), writeln(Zensko1),

nl,

assert(zaruceni(Musko1, Zensko2)),%dodaje se novi par

assert(zaruceni(Musko2, Zensko1)),%dodaje se novi par

zaruci.

**%Sve dok ima samaca se ponavlja uparivanje samaca i zarucivanje**

zaruci:-

not(samac(\_)).

### **%Predikat koji potvrđuje konacne parove**

vjencaj:-

```
zaruceni(Musko, Zensko),  
retract(zaruceni(Musko, Zensko)),%micu se zaruceni parovi  
assert(vjencani(Musko, Zensko)),%dodaju se vjencani parovi  
write('\t'), write(Musko),write(' i '),write(Zensko), nl,  
vjencaj.
```

### **%Prekida se izvodjenje kada vjencamo sve zarucene parove**

vjencaj:-

```
not(zaruceni(_,_)).
```

### **%Predikat koji omogućuje upis vlastitih preferenci**

unos:-

```
write('Unesite naziv datoteke: '), read(Datoteka),%uzima se naziv datoteke  
nl,  
write('Unijeti broj parova: '), read(Broj_parova),%uzima se broj parova  
assert(parovi(Broj_parova)),%zapisuje se koliko imamo parova  
nl,  
string_concat(Datoteka, '.txt', Datoteka2),%dodajemo ekstenziju za datoteku  
open(Datoteka2, append, S),%otvaramo novokreiranu datoteku i dodajemo podatke u nju  
write(S,Broj_parova),%zapisujemo broj parova u datoteku  
write(S,'\n'),%novi red  
writeln('Unesite muskarce i njihove preference: '),  
zapisi_podatke(Broj_parova, S),%poziva se predikat koji upisuje muskarce i njihove  
preference  
nl,  
writeln('Unesite zene i njihove preference: '),  
zapisi_podatke(Broj_parova, S),%poziva se predikat koji upisuje zene i njihove preference  
close(S),%zatvara se tok podataka  
nl,  
write('Vasi podaci su zapisani u datoteku '),  
writeln(Datoteka2),  
nastavak_programa.%poziva se predikat za jednostavan nastavak programa
```

```
zapisi_podatke(0,_).
```

**%Predikat koji zapisuje osobe i njihove preference u datoteku**

```
zapisi_podatke(Broj_parova, S):-
```

```
    write('Osoba i preference: '),
```

```
    read_line_to_codes(user_input, Input),%cita se korisnikov upis
```

```
    string_to_atom(Input,IA),%unos pretvaramo u atom
```

```
    write(S,IA),%zapisujemo unos
```

```
    write(S,'\n'),%nova linija
```

```
    Broj_parova2 is Broj_parova-1,%smanjujemo kako bi unijeli tocan broj osoba i preferenci
```

**(kao za for petlju)**

```
    zapisi_podatke(Broj_parova2, S).
```

**%Predikat koji ispisuje pomoc, odnosno upute za korištenje programa i neke bitne informacije koje korisnik treba znati kako bi ucinkovito koristio program**

```
hlp:-
```

```
    nl, writeln('~~~~~UPUTSTVA ZA KORISTENJE PROGRAMA~~~~~'), nl,
```

```
    writeln('-----'),
```

```
    writeln('Kratak opis:'), nl,
```

```
    writeln('Program je namijenjen rjesavanju problema stabilnih brakova'),
```

```
    writeln('Cilj je upariti N broj muskaraca i N broj zena prema njihovim preferencama, '),
```

```
    writeln('kako bi svaki od njih imao prikladnog para. Princip uparivanja je ostvaren '),
```

```
    writeln('pomocu Gale i Shapley algoritma kojem je cilj pokazati kako se iz bilo '),
```

```
    writeln('kojeg skupa preferenci mogu ostvariti stabilni brakovi.').
```

```
    writeln('-----'), nl,
```

```
    writeln('Neke stvari koje je potrebno znati(Ogranicenja programa i nacin unosa podataka)'),
```

```
    nl,
```

```
    writeln('1. Odnos preferenci: '),
```

```
    write('\t'), writeln('Najpozeljniji/a -----> Najnepoželjniji/a'),
```

```
    write('\t'), writeln('  Lijevo ----->   Desno   '), nl,
```

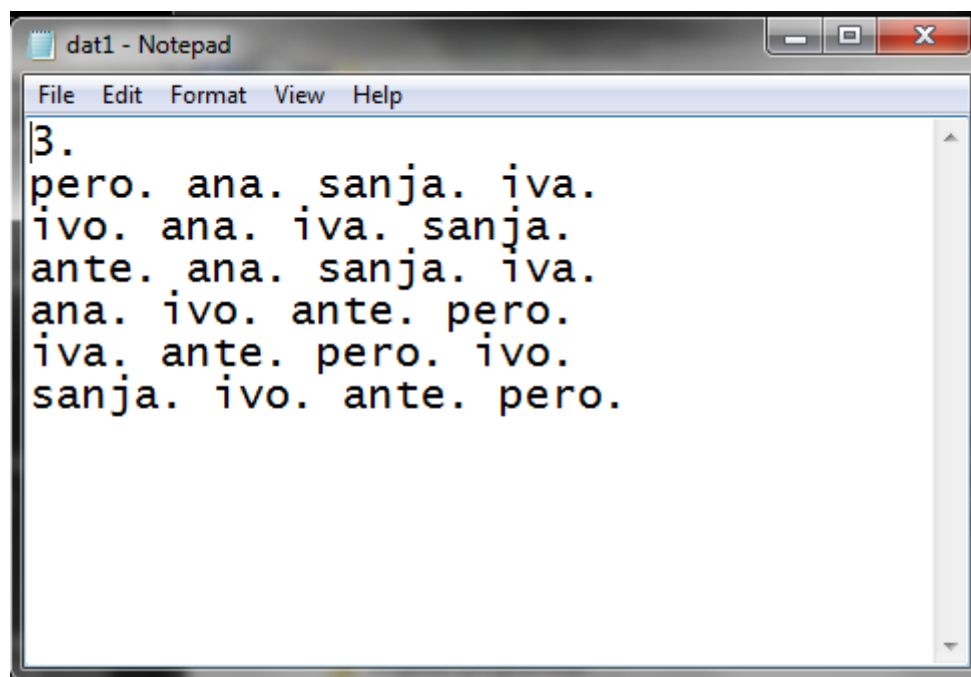
```
    writeln('2. Ucitavanje datoteke: '),
```

```
    write('\t'), writeln('Potrebno je upisati naziv zajedno sa ekstenzijom unutar jednostrukih  
navodnika.').
```

```
    write('\t'), writeln('  npr. "test.txt"'), nl,
```

```
writeln('3. Unos osobe i njenih preferenci: '),
      write('\t'), writeln('Prvo se unosi osoba za koju zelimo dodati preference, a zatim slijede
preference '),
      write('\t'), writeln('redom od najpozeljnijeg partnera do najnepoželjnijeg partnera. Pritom je
potrebno '),
      write('\t'), writeln('nakon svakog imena staviti tocku.'),
      write('\t'), writeln('    npr. pero. ana. sanja. iva.'),
      writeln('-----'),
      nl,
      writeln('BITNA NAPOMENA!'),
      writeln('DA BI PROGRAM RADIO KAKO TREBA POTREBAN JE JEDNAK BROJ MUSKARACA
I ZENA!'),
      nastavak_programa.%poziv predikata koji definira nastavak izvođenja programa
```

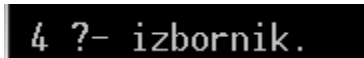
Slijedi također prikaz datoteke s podacima kako bi se vidjelo na koji način su formatirani podaci koji se koriste za rješavanje problema stabilnih brakova.



Slika 4.1. Prikaz formata datoteke s podacima

## 5. Prikaz primjera izvođenja programa

Program se pokreće pozivom predikata **izbornik/0**.



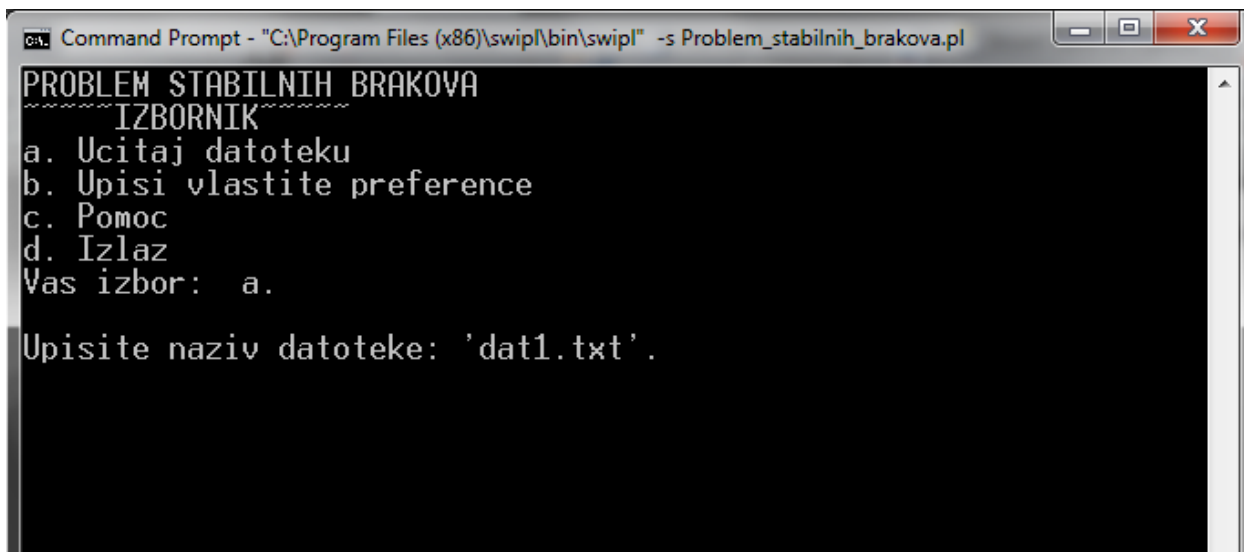
```
4 ?- izbornik.
```

Slika 5.1. Pokretanje programa

Ispisuje se zatim izbornik koji korisniku nudi nekoliko funkcionalnosti:

- a. Učitaj datoteku
- b. Upiši vlastite podatke
- c. Pomoć
- d. Izlaz

U prvom slučaju odabiremo prvu funkcionalnost, odnosno učitavanje datoteke. Datoteka koju ćemo učitati ima ime 'dat1.txt', te sadrži iste podatke kao i prikaz formata datoteke u prethodnom poglavlju.

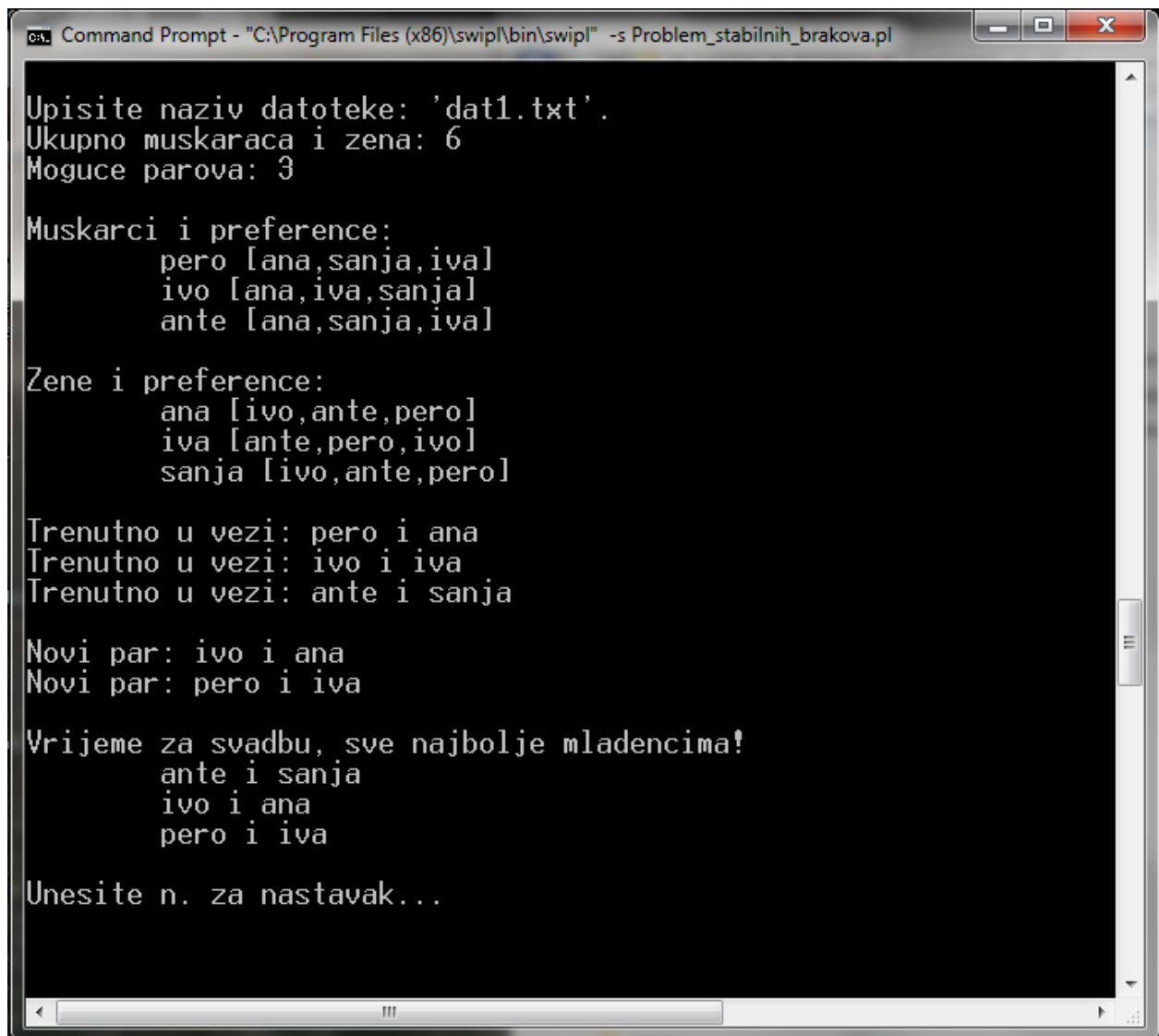


```
cmd - Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl
PROBLEM STABILNIH BRAKOVA
~~~~~IZBORNIK~~~~~
a. Ucitaj datoteku
b. Upisi vlastite preference
c. Pomoc
d. Izlaz
Vas izbor: a.
Upisite naziv datoteke: 'dat1.txt'.
```

Slika 5.2. Upis odabrane datoteke

Po unosu imena odabrane datoteke slijedi proces rješavanja problema stabilnih brakova, te se tokom izvođenja korisniku ispisuju bitne informacije kako bi korisnik mogao pratiti što se događa i do kojih je promjena došlo.





```
Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl

Upisite naziv datoteke: 'dat1.txt'.
Ukupno muskaraca i zena: 6
Moguće parova: 3

Muskarci i preference:
    pero [ana,sanja,iva]
    ivo [ana,iva,sanja]
    ante [ana,sanja,iva]

Zene i preference:
    ana [ivo,ante,pero]
    iva [ante,pero,ivo]
    sanja [ivo,ante,pero]

Trenutno u vezi: pero i ana
Trenutno u vezi: ivo i iva
Trenutno u vezi: ante i sanja

Novi par: ivo i ana
Novi par: pero i iva

Vrijeme za svadbu, sve najbolje mladencima!
    ante i sanja
    ivo i ana
    pero i iva

Unesite n. za nastavak...
```

Slika 5.3. Proces rješavanja stabilnih brakova i ispis kod ključnih točaka

Korisnik zatim po povratku na izbornik može ponovno odabrati prvu funkcionalnost i upisati naziv neke druge datoteke za koju želi riješiti problem stabilnih brakova.

Prvo se ispisuju podaci koji se nalaze u datoteci kako bi korisnik vidio koliko ukupno osoba ima, koliko je moguće parova dobiti, te za svaku osobu je prikazano njeno ime i lista preferenci.

```
Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl

PROBLEM STABILNIH BRAKOVA
~~~~~
IZBORNIK
a. Učitaj datoteku
b. Upisi vlastite preference
c. Pomoc
d. Izlaz
Vas izbor: a.

Upisite naziv datoteke: 'dat3.txt'.
Ukupno muskaraca i zena: 20
Moguće parova: 10

Muskarci i preference:
abe [abi,eve,cath,ivy,jan,dee,fay,bea,hope,gay]
bob [cath,hope,abi,dee,eve,fay,bea,jan,ivy,gay]
col [hope,eve,abi,dee,bea,fay,ivy,gay,cath,jan]
dan [ivy,fay,dee,gay,hope,eve,jan,bea,cath,abi]
ed [jan,dee,bea,cath,fay,eve,abi,ivy,hope,gay]
fred [bea,abi,dee,gay,eve,ivy,cath,jan,hope,fay]
gav [gay,eve,ivy,bea,cath,abi,dee,hope,jan,fay]
hal [abi,eve,hope,fay,ivy,cath,jan,bea,gay,dee]
ian [hope,cath,dee,gay,bea,abi,fay,ivy,jan,eve]
jon [abi,fay,jan,gay,eve,bea,dee,cath,ivy,hope]

Zene i preference:
abi [bob,fred,jon,gav,ian,abe,dan,ed,col,hal]
bea [bob,abe,col,fred,gav,dan,ian,ed,jon,hal]
cath [fred,bob,ed,gav,hal,col,ian,abe,dan,jon]
dee [fred,jon,col,abe,ian,hal,gav,dan,bob,ed]
eve [jon,hal,fred,dan,abe,gav,col,ed,ian,bob]
fay [bob,abe,ed,ian,jon,dan,fred,gav,col,hal]
gay [jon,gav,hal,fred,bob,abe,col,ed,dan,ian]
hope [gav,jon,bob,abe,ian,dan,hal,ed,col,fred]
ivy [ian,col,hal,gav,fred,bob,abe,ed,jon,dan]
jan [ed,hal,gav,abe,bob,jon,col,ian,fred,dan]
```

Slika 5.4. Ispis podataka pročitanih iz datoteke

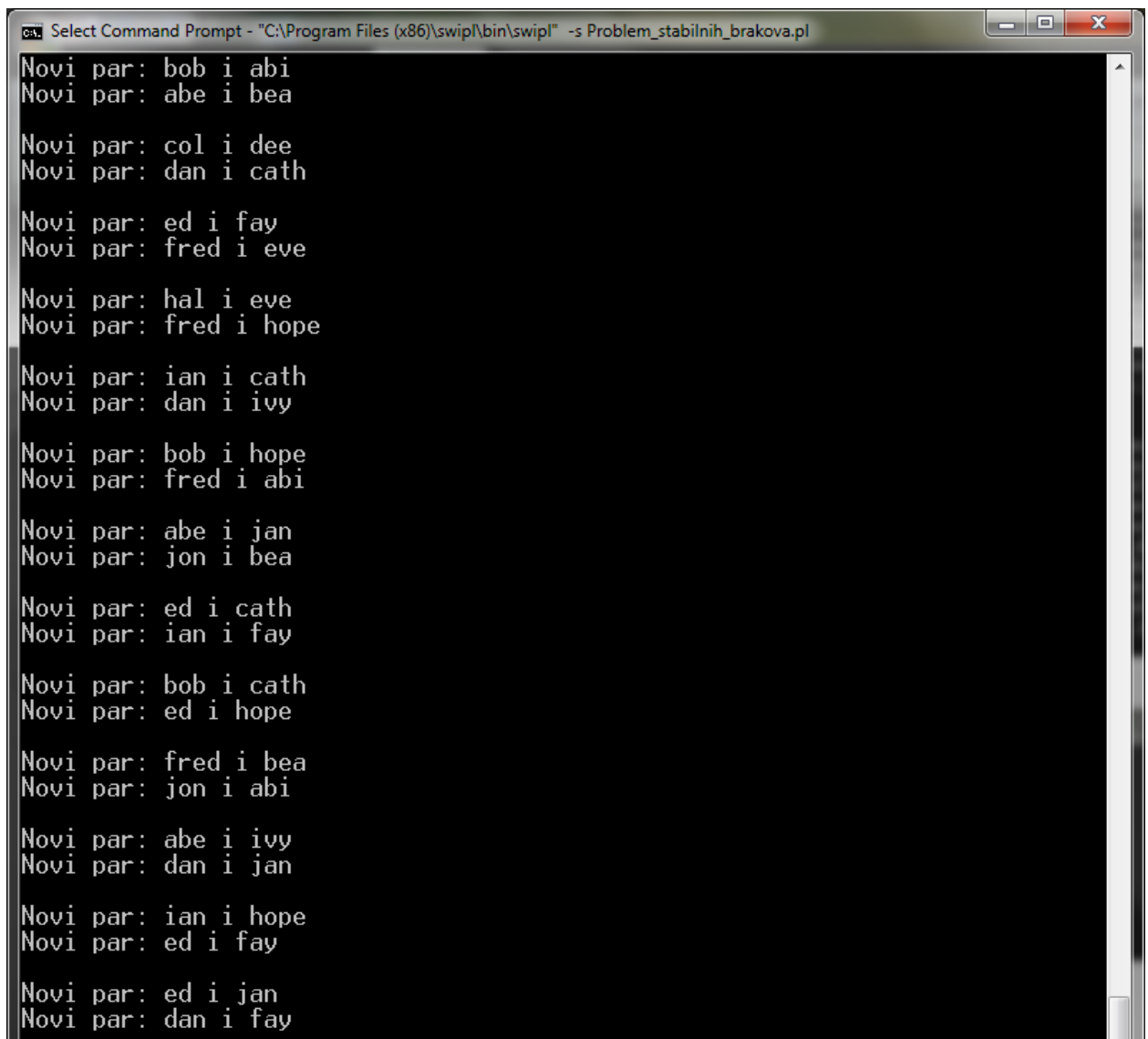
Slijedeće se korisniku prikazuje kako je protekao prvi korak rješavanja problema stabilnih brakova, odnosno ispisuju se početni parovi dobiveni prema podacima.

```
Select Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl

Trenutno u vezi: abe i abi
Trenutno u vezi: bob i bea
Trenutno u vezi: col i cath
Trenutno u vezi: dan i dee
Trenutno u vezi: ed i eve
Trenutno u vezi: fred i fay
Trenutno u vezi: gav i gay
Trenutno u vezi: hal i hope
Trenutno u vezi: ian i ivy
Trenutno u vezi: jon i jan
```

Slika 5.5. Rezultati početnog uparivanja

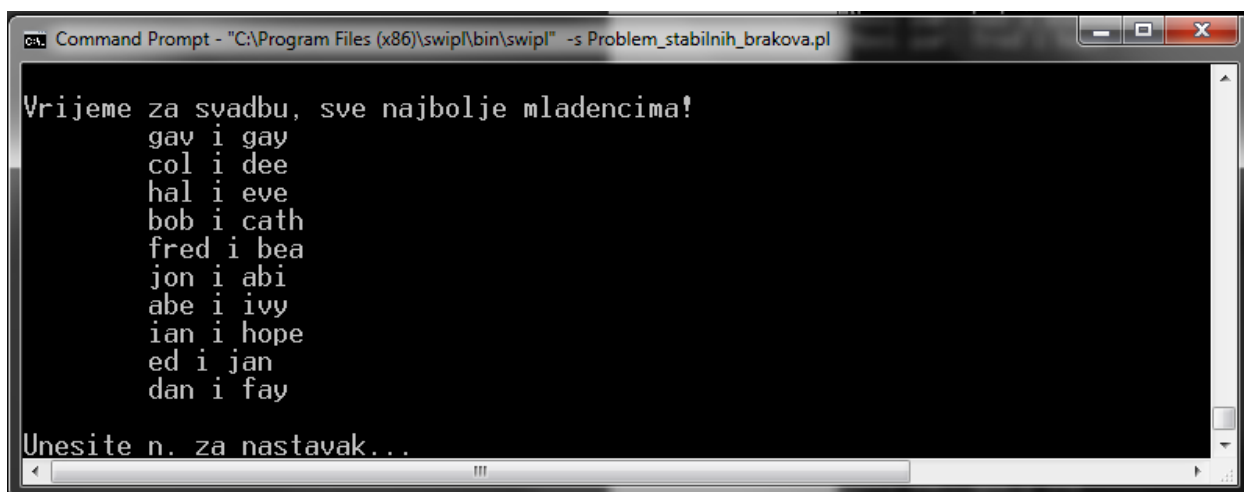
Nakon što se provede početno uparivanje prelazi se na drugi korak rješavanja problema stabilnih brakova, gdje se provjerava za svakog muškarca da li postoji žena koju on više preferira u usporedbi sa svojom trenutnom partnericom. Ukoliko je to istina, te ujedno i žena koju on preferira isto više preferira njega u usporedbi sa njenim trenutnim partnerom oboje ostavljaju svoje zaručnike i ulaze u vezu, a ostavljeni budu zaručeni jedan za drugog. Korisniku se tokom tog procesa ispisuju novonastali parovi.



```
Select Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl
Novi par: bob i abi
Novi par: abe i bea
Novi par: col i dee
Novi par: dan i cath
Novi par: ed i fay
Novi par: fred i eve
Novi par: hal i eve
Novi par: fred i hope
Novi par: ian i cath
Novi par: dan i ivy
Novi par: bob i hope
Novi par: fred i abi
Novi par: abe i jan
Novi par: jon i bea
Novi par: ed i cath
Novi par: ian i fay
Novi par: bob i cath
Novi par: ed i hope
Novi par: fred i bea
Novi par: jon i abi
Novi par: abe i ivy
Novi par: dan i jan
Novi par: ian i hope
Novi par: ed i fay
Novi par: ed i jan
Novi par: dan i fay
```

Slika 5.6. Proces izvođenja drugog koraka, prikaz novih parova

Nakon što više ne postoje muškarac i žena koji se međusobno više preferiraju u usporedbi s njihovim partnerima uparivanje je završilo te je vrijeme za svadbu. Korisniku se ispisuju konačni parovi.



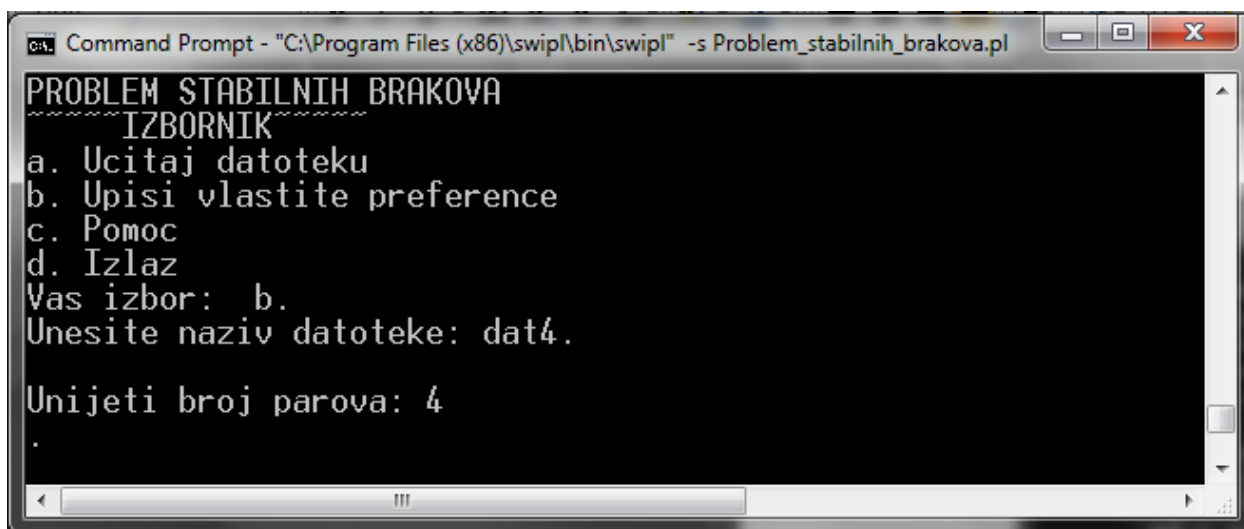
```
Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl

Vrijeme za svadbu, sve najbolje mladencima!
  gav i gay
  col i dee
  hal i eve
  bob i cath
  fred i bea
  jon i abi
  abe i ivy
  ian i hope
  ed i jan
  dan i fay

Unesite n. za nastavak...
```

Slika 5.7. Konačni parovi i vjenčanje

Korisnik također može odabrati mogućnost unosa nekih vlastitih podataka umjesto korištenja već postojećih datoteka. U tom slučaju će program od korisnika prvo tražiti da unese naziv datoteke u koju će se spremi podaci koje će unijeti. Nakon toga program traži da korisnik unese koliko će parova biti uneseno u datoteku.



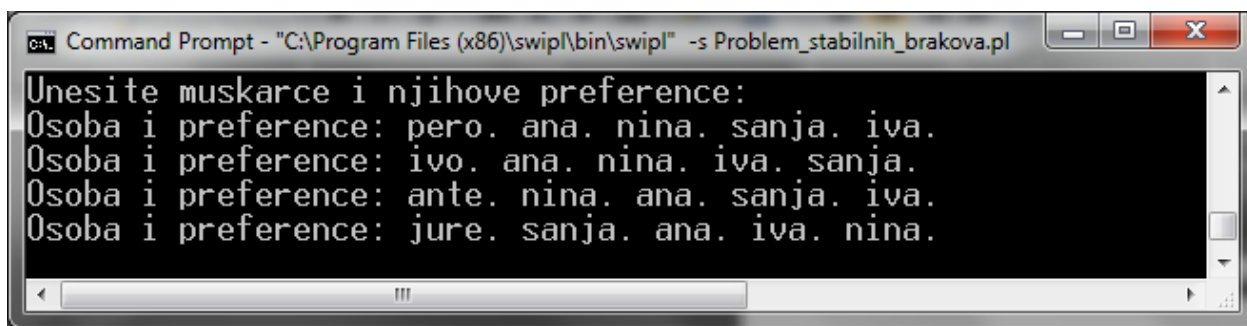
```
Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl

PROBLEM STABILNIH BRAKOVA
~~~~~IZBORNIK~~~~~
a. Učitaj datoteku
b. Upisi vlastite preference
c. Pomoc
d. Izlaz
Vas izbor: b.
Unesite naziv datoteke: dat4.

Unijeti broj parova: 4
.
```

Slika 5.8. Upisivanje vlastitih podataka: naziv datoteke i broj parova

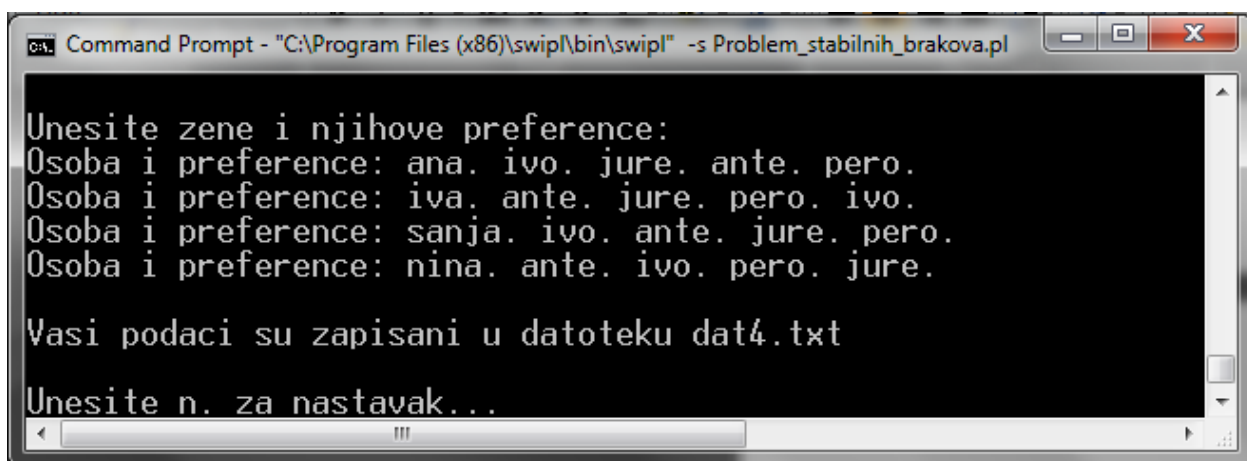
Nakon što korisnik unese tražene početne podatke program će ga za svaku osobu, počevši od muškaraca, tražiti da unese ime osobe i imena osoba koje preferiraju redom od najpoželjnije do najnepoželjnije.



```
C:\Program Files (x86)\swipl\bin\swipl -s Problem_stabilnih_brakova.pl

Unesite muskarce i njihove preference:
Osoba i preference: pero. ana. nina. sanja. iva.
Osoba i preference: ivo. ana. nina. iva. sanja.
Osoba i preference: ante. nina. ana. sanja. iva.
Osoba i preference: jure. sanja. ana. iva. nina.
```

Slika 5.9. Unos podataka za muškarce



```
C:\Program Files (x86)\swipl\bin\swipl -s Problem_stabilnih_brakova.pl

Unesite zene i njihove preference:
Osoba i preference: ana. ivo. jure. ante. pero.
Osoba i preference: iva. ante. jure. pero. ivo.
Osoba i preference: sanja. ivo. ante. jure. pero.
Osoba i preference: nina. ante. ivo. pero. jure.

Vasi podaci su zapisani u datoteku dat4.txt

Unesite n. za nastavak...
```

Slika 5.10. Unos podataka za žene

Nakon što korisnik unese sve podatke program će korisniku javiti kako su podaci zapisani u datoteku koju je naveo. Korisnik zatim može odabirom prve funkcionalnosti učitati vlastitu datoteku i vidjeti rezultate rješavanja problema stabilnih brakova.

Posljednja funkcionalnost je ispis pomoći koja korisniku ispisuje informacije bitne za izvođenje programa, koja ograničenja program sadrži ta kako unositi podatke i komande.

```
Command Prompt - "C:\Program Files (x86)\swipl\bin\swipl" -s Problem_stabilnih_brakova.pl

PROBLEM STABILNIH BRAKOVA
~~~~~IZBORNIK~~~~~
a. Ucitaj datoteku
b. Upisi vlastite preference
c. Pomoc
d. Izlaz
Vas izbor: c.

~~~~~UPUTSTVA ZA KORISTENJE PROGRAMA~~~~~
-----
Kratak opis:

Program je namijenjen rjesavanju problema stabilnih brakova
Cilj je upariti N broj muskaraca i N broj zena prema njihovim preferencama,
kako bi svaki od njih imao prikladnog para. Princip uparivanja je ostvaren
pomocu Gale i Shapley algoritma kojem je cilj pokazati kako se iz bilo
kojeg skupa preferenci mogu ostvariti stabilni brakovi.
-----

Neke stvari koje je potrebno znati(0granicenja programa i nacin unosa podataka)

1. Odnos preferenci:
    Najpozeljniji/a -----> Najnepoželjniji/a
    Lijevo -----> Desno

2. Ucitavanje datoteke:
    Potrebno je upisati naziv zajedno sa ekstenzijom unutar jednostrukih navodnika.
    npr. 'test.txt'

3. Unos osobe i njenih preferenci:
    Prvo se unosi osoba za koju zelimo dodati preference, a zatim slijede preference
    redom od najpoželjnijeg partnera do najnepoželjnijeg partnera. Pritom je potrebno
    nakon svakog imena staviti tocku.
    npr. pero. ana. sanja. iva.
    -----

BITNA NAPOMENA!
DA BI PROGRAM RADIO KAKO TREBA POTREBAN JE JEDNAK BROJ MUSKARACA I ZENA!

Unesite n. za nastavak...
```

Slika 5.11. Funkcionalnost Pomoć

## 6. Zaključak

Problem koji se rješavao u projektu je jedan od uobičajenih problema kojima se bavi informatika i algoritmi. U principu se može primijeniti na mnogo primjera (ne samo za brakove) gdje je potrebno spojiti dva suprotna skupa pritom poštivati zadana ograničenja. Sličan problem sam teorijski obrađivala u sklopu seminarskog rada na kolegiju Inteligentni sustavi koji se bavio problemom raspoređivanja medicinskih sestara u raspored smjena prema određenim ograničenjima koja su uključivala ona pravne i radne prirode, te ona osobne prirode.

Proučavajući izvore koji su se nudili i načine na koje bih mogla implementirati i riješiti problem naišla sam na mnogo različitih rješenja implementiranih u različitim programskim jezicima. Protezali su se od vrlo jednostavnih implementacija u C jeziku do čak poprilično kompliciranih u samom Prologu.

Budući da nisam jako vješta u korištenju Prolog jezika odlučila sam se na implementaciju pomoću Gale-Shapely algoritma na vrlo jednostavan način imitirajući for petlje i korištenjem listi te jednostavnom manipulacijom istih. Odlučila sam malo proširiti mogućnosti samog programa na način da nema već unaprijed određene činjenice, pa prema tome i samo jednu verziju izvođenja. Stoga sam koristila jednostavno učitavanje podataka iz unaprijed strukturirane datoteke kojih u principu možemo imati mnogo sa različitim podacima. Također je omogućen vlastiti unos podataka na način da se oni zapisuju u datoteku kako bi se lakše mogao kontrolirati unos korisnika, a i da se iskoristi već implementirana mogućnost čitanja podataka iz datoteke.

Korisničko sučelje je jednostavno, te ispisuje ključne informacije korisniku kako bi mogao vidjeti što se događa tokom izvođenja programa.

Program bi se sigurno mogao proširiti boljim sučeljem i drugačijim unosom podataka, no to je bilo malo iznad mojih sposobnosti sa prologom pa to ostavljam za neki drugi puta.

## 7. Literatura

1. Wirth, N., 1976., Algorithms + data structures = programs, Zurich, Switzerland
2. Stable marriage problem, dostupno na:
3. [http://rosettacode.org/wiki/Stable\\_marriage\\_problem](http://rosettacode.org/wiki/Stable_marriage_problem)
4. Gusfield, D., Irving, R. W., -, The stable Marriage Problem, The MIT Press, dostupno na:  
<http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15251-f10/Site/Materials/Lectures/Lecture21/lecture21.pdf>
5. SWI Prolog Manual, dostupno na: <http://www.swi-prolog.org/pldoc/refman/>