**FAKULTET ORGANIZACIONIH NAUKA**

**UNIVERZITET U BEOGRADU**

**DVOSTRUKO SPREGNUTA LISTA U ATP FORMI REALIZOVANA PREKO DATOTEKE**

Seminarski rad iz predmeta Programiranje 1

Profesor: Student:

Dr. Saša D. Lazarević Vladimir Lazarević 2020/0121

Beograd, Jul 2021.

Contents

[UVOD 3](#_Toc76996548)

[FUNKCIJE NAD LISTOM 4](#_Toc76996549)

[1) ubaci() 4](#_Toc76996550)

[2) izbaci() 4](#_Toc76996551)

[3) promeni() 5](#_Toc76996552)

[4) prikazi() 6](#_Toc76996553)

[5) nadji() 6](#_Toc76996554)

[6) sortiraj() 7](#_Toc76996555)

[KORISNIČKI MENI 8](#_Toc76996556)

[ZAKLJUČAK 9](#_Toc76996557)

[LITERATURA 10](#_Toc76996558)

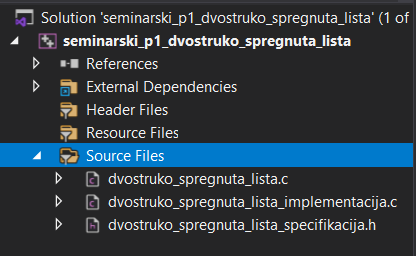
# UVOD

Lista može biti realizovana korišćenjem pokazivača, niza ili datoteke. Tema ovog seminarskog je realizacija dvostruko spregnute liste u ATP formi preko datoteke. Koristiću binarnu datoteku “lista.dat” u kojoj će se nalaziti lista. Funkcije koje će moći da se izvršavaju nad listom su: ubaci, izbaci, promeni, prikaži, nađi i sortiraj. Tim funkcijama će moći da se pristupi preko korisničkog menija.

U okviru projekta postojaće 3 fajla:

1. dvostruko\_spregnuta\_lista.c
2. dvostruko\_spregnuta\_lista\_specifikacija.h
3. dvostruko\_spregnuta\_lista\_implementacija.c

* U fajlu dvostruko\_spregnuta\_lista.c će se nalaziti opis projekta i funkcija main u kojoj će biti pozvana funkcija korisnički meni.
* U fajlu dvostruko\_spregnuta\_lista\_specifikacija.h će se nalaziti prototipi funkcija: ubaci, izbaci, promeni, prikaži, nađi, sortiraj i korisnički meni.
* U fajlu dvostruko\_spregnuta\_lista\_implementacija.c će se nalaziti logika funkcija koje se nalaze u specifikaciji.



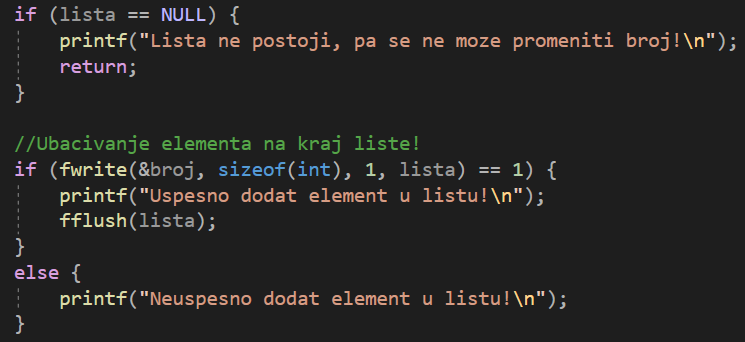
# FUNKCIJE NAD LISTOM

## ubaci()

Funkcija ubaci ubacuje određeni broj u listu. Njen prototip izgleda ovako:



Njen kod izgleda ovako:



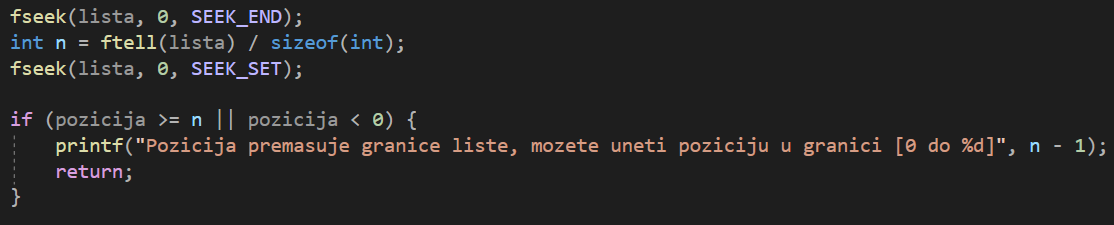
Prvo proveravam da li postoji lista, ovaj deo koda ću zanemariti u opisu ostalih funkcija. Kad radimo sa ovakvim listama, dovoljno je da samo koristimo funkciju fwrite koja kao parameter prima broj koji se dodaje u listu, njegovu velicinu u bajtovima, koliko se elemenata upisuje i u koju datoteku se upisuje. Ako se izvršila funkcija fwrite pozivamo funkciju fflush. Fflush vraća 0 ako je upis u datoteku izvršen kako treba, a vraća EOF ako se desio neki problem.

## izbaci()

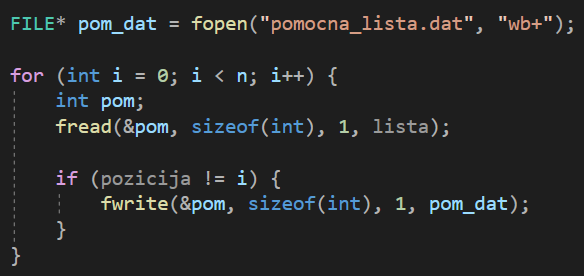
Funkcija izbaci izbacuje element sa određene pozicije u listi. Njen prototip izgleda ovako:

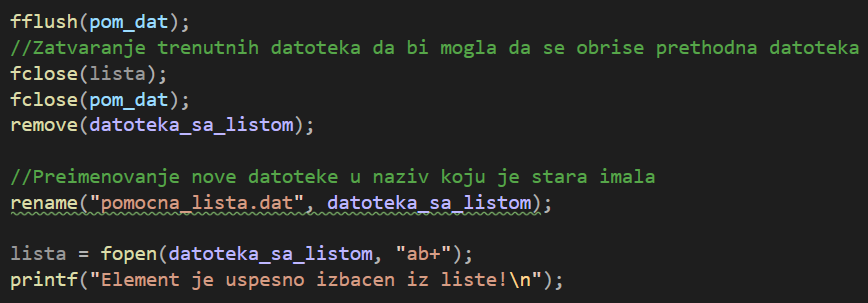


Prvo određujemo broj elemenata listen a sledeći način i utvrđujemo da li je pozicija u granicama liste.



Zatim pravimo pomoćnu datoteku jer ne dozvoljava se brisanje određenog člana iz datoteke(<https://stackoverflow.com/questions/19714717/remove-bytes-from-a-file-c>), i upisujemo sve članove iz originalne datoteke koji se ne nalaze na traženoj poziciji:





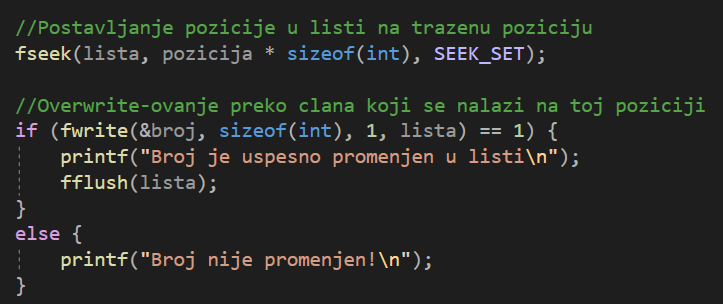
Moramo zatvoriti obe datoteke kako bi mogli da obrisemo staru I preimenujemo novu u isti naziv koji ima stara.

## promeni()

Funkcija promeni menja element sa određene pocije u listi sa zadatim brojem. Njen prototip izgleda ovako:



Moramo postaviti poziciju na traženi element u datoteci i onda napisati odnosno overwite-ovati zadati broj preko tog broja.

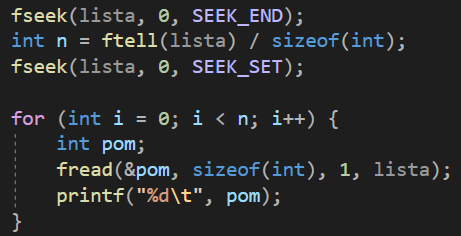


## prikazi()

Funkcija prikazi prikazuje sadržaj liste. Njen prototip izgleda ovako:



Njena logika izgleda ovako:



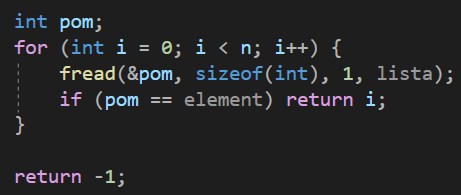
Krećemo se kroz datoteku i čitamo članove u promenljivu pom i prikazujemo je.

## nadji()

Funkcija vraća indeks pozicije traženog broja(ali prvog pojavljivanja). Njen protip izgleda ovako:



Logika:



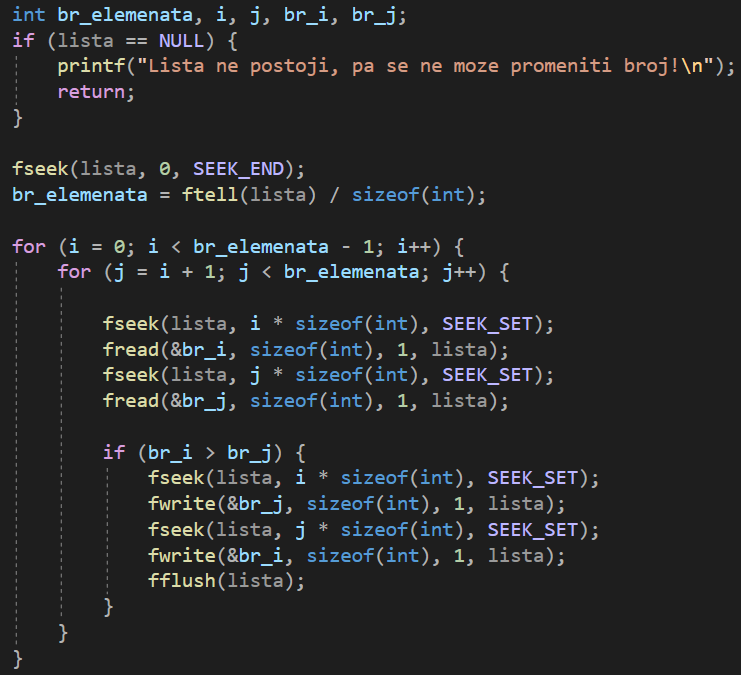
Krećemo se kroz for petlju i ako nađemo član koji se poklapa sa trenutno posmatranim članom liste vraćamo njegovu poziciju. Ako ne postoji taj član vraćamo -1.

## sortiraj()

Funkcija sortiraj sortira listu u rastućem redosledu i njen prototip izgleda ovako:

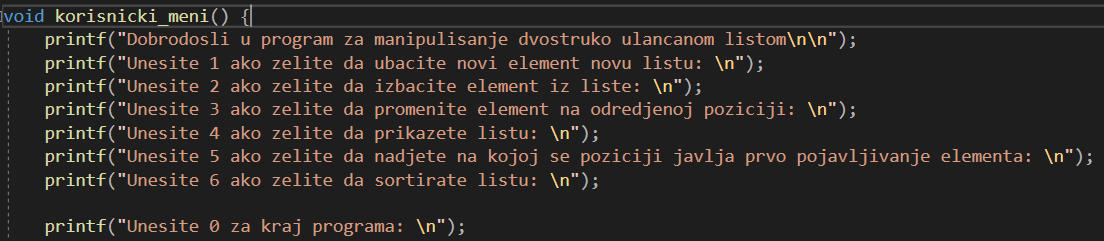


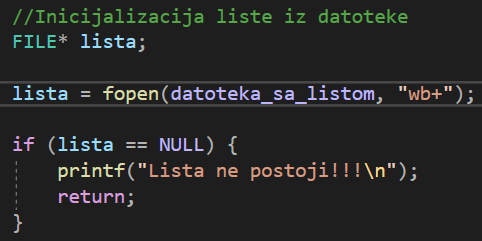
Logika:



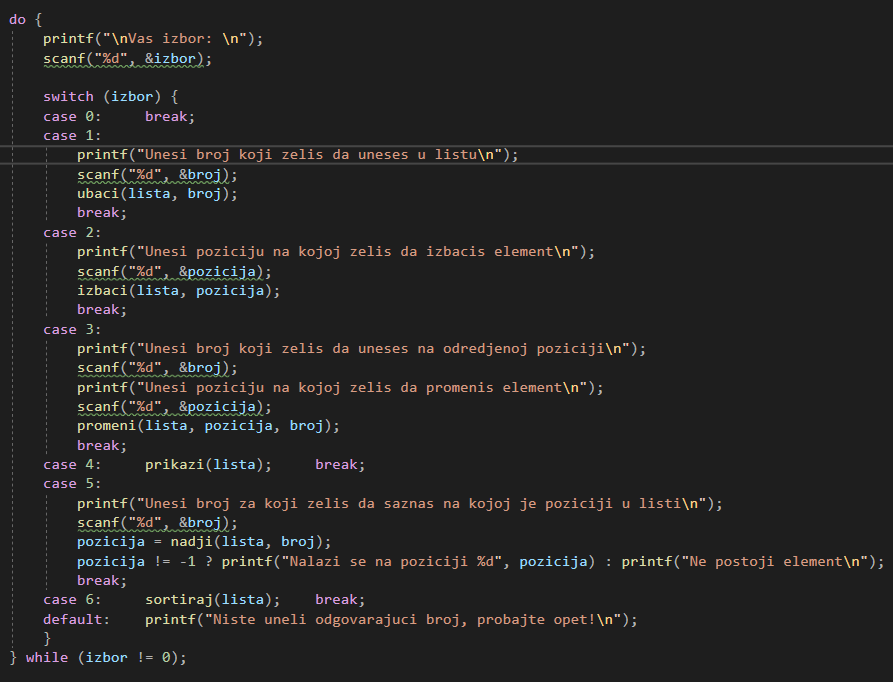
Na početku postavljamo datotečni pokazivač na kraj datoteke kako bi pomoću funkcije ftell dobili broj elemenata liste. Nakon toga primenjujemo selection sort gde pomeramo datotečni pokazivač funkcijom fseek i poredimo taj element sa određenim fiksiranim elementom.

# KORISNIČKI MENI





Inicijalizujemo binarnu datoteku



Za određeni izbor korisniku će se tražiti dodatni podaci koji su potrebni da bi se funkcija izvršila. Za izbor 0 program se prekida.

# ZAKLJUČAK

Shvatio sam da je jako bolje raditi na ovakav način sa listama jer postaju perzistentne, a pri tome jako je lakše manipulisati njima. Problem je samo što se u debugeru ne može tačno videti gde se koji član liste nalazi, što je dosta lakše kada se radi preko pokazivača.

# LITERATURA

1. [Програмирање 1 – Збирка задатака (други део)](http://silab.fon.bg.ac.rs/wp-content/uploads/2019/05/P1_Zbirka_deo02_v1.pdf)
2. Fflush opis - <https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/fflush?view=msvc-160>
3. Fwrite opis - <https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/fwrite?view=msvc-160>
4. Fseek opis - <https://www.tutorialspoint.com/c_standard_library/c_function_fseek.htm>
5. Fread opis - <https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/fread?view=msvc-160>
6. Sklanjanje elementa iz liste - <https://stackoverflow.com/questions/19714717/remove-bytes-from-a-file-c>