

Baza zadataka za ispit:
Programiranje II
Matematički fakultet u Beogradu

30. maj 2010

Sadržaj

1	Obnavljanje gradiva Programiranja 1	2
1.1	Lakši zadaci	2
1.2	Srednje teški zadaci	2
1.3	Teži zadaci	2
2	Datoteke, argumenti komandne linije.	2
2.1	Lakši zadaci	2
2.2	Srednje teški zadaci	4
3	Kompilacija iz više izvornih datoteka	5
3.1	Srednje teški zadaci	5
3.2	Teži zadaci	5
4	Bitski operatori	6
4.1	Lakši zadaci	6
4.2	Srednje teški zadaci	7
4.3	Teži zadaci	7
5	Sortiranje i pretraga	8
5.1	Lakši zadaci	8
5.2	Srednje teški zadaci	8
5.3	Teži zadaci	9
6	Rekurzija	10
6.1	Lakši zadaci	10
6.2	Srednje teški zadaci	10
6.3	Teži zadaci	11
7	Rekurzivne implementacije algoritama pretrage i sortiranja	11
7.1	Lakši zadaci	11
7.2	Srednje teški zadaci	12
8	Pokazivači na funkcije	12
8.1	Lakši zadaci	12
8.2	Srednje teški zadaci	12
9	Dinamička alokacija memorije	12
9.1	Lakši zadaci	12
9.2	Srednje teški zadaci	12
10	Višedimenzioni nizovi i matrice	13
10.1	Lakši zadaci	13
10.2	Srednje teški zadaci	14

11 Funkcije lfind, lsearch, bsearch i qsort	15
11.1 Lakši zadaci	15
11.2 Srednje teški zadaci	15
12 Liste.	15
12.1 Lakši zadaci	15
12.2 Srednje teški zadaci	16
13 Binarna stabla.	17
13.1 Lakši zadaci	17
13.2 Srednje teški zadaci	17
14 Stekovi, redovi.	18
14.1 Lakši zadaci	18
14.2 Srednje teški zadaci	18

1 Obnavljanje gradiva Programiranja 1

1.1 Lakši zadaci

1. Napisati program koji računa razliku $S1 - S2$, gde je $S1$ suma elemenata sa parnim indeksima u nizu A , a $S2$ je suma elemenata sa neparnim indeksima u nizu A .
2. Napisati program koji određuje dva najveća elementa niza A .
3. Napisati program koji za dati string s broji sve pojave datog slova l u njemu. Broje se pojavljivanja slova l i u obliku malog i u obliku velikog slova. Npr: za string "abA" i slovo 'a' rezultat treba da bude 2.
4. Napisati program koji ispituje da li je dati string palindrom.
5. Tačka u ravni se zadaje svojim koordinatama (x,y) . Predstaviti je strukturom i uvesti ime tipa Tacka. Napisati C program koji učitava niz tačaka i pronalazi onu koja je najbliža koordinatnom početku. Na standardni izlaz ispisati nađenu tačku i njeno rastojanje.
6. Napisati funkciju koja kvadrira one elemente niza koji su veći od aritmetičke sredine niza.

1.2 Srednje teški zadaci

1. Napisati program koji iz niza A izbacuje sve elemente $A[i]$ koji imaju sledeće svojstvo: $i \geq 0$ i $A[i]$ je manji od svih elemenata koji mu prethode u nizu A .
2. Napisati program koji prikazuje prvih n Fibonačijevih stringova:

```
F[0] = "a"
F[1] = "b"
F[i] = F[i-1] + F[i-2], i > 1
```

3. Sa standardnog ulaza se učitava niz od n ($n \leq 100$) tačaka u ravni takvih da nikoje tri tačke nisu kolinearne. Tačke se zadaju parom svojih koordinata. Ispitati da li taj niz tačaka određuje konveksni mnogougao i rezultat ispisati na standardni izlaz.
4. Napisati C program koji će učitati linije sa ne više od 80 karaktera sa std ulaza i prepisati ih na std izlaz tako da se uklone sva slova engleske abecede kojih nema u srpskoj azbuci. Učitavanje linije sa standardnog ulaza realizovati pozivanjem funkcije

```
int getline(char *s, int max)
```

Funkcija učitava karakter po karakter sa std ulaza i smešta u nisku s , sve dok ne pročita EOF, $\backslash n$ ili se pročita maksimalan broj karaktera u liniji (zadat sa 2. argumentom - `max`). Funkcija vraća dužinu pročitane linije s .

5. Napisati funkciju koja za dati niz M dužine N i interval $[a,b]$ vraća indekse minimalnog i maksimalnog elementa iz traženog interavala.

1.3 Teži zadaci

1. Napisati program koji ispituje da li je dati niz A periodičan.
2. Napisati program koji za dati string koji predstavlja broj u rimskom zapisu određuje vrednost tog celog broja.
3. Napisati program koji ispituje da li dati string $s1$ nastaje cikličnim pomeranjem drugog datog stringa $s2$. Npr: za $s1="abcde"$ i $s2="cdeab"$ $s1$ nastaje ciklicnom rotacijom u desno stringa $s2$ za 2 mesta.

2 Datoteke, argumenti komandne linije.

2.1 Lakši zadaci

1. Napisati program koji prepisuje datoteku `ulaz.txt` u datoteku `izlaz.txt` i to
 - (a) karakter po karakter
 - (b) liniju po liniju.

(Povezati sa ranije viđenim programima koji su prepisivali standardni ulaz na standardni izlaz. Diskutovati i standardni izlaz za greške (`stderr`) i sve greške ubuduće ispisivati isključivo na njega.)

2. Napisati program koji ispisuje broj navedenih argumenata komandne linije, a zatim i same argumenate i njihove redne brojeve. Npr. ukoliko se program pokrene sa

```
./echoargs Ovo je primer. 123 "Kako ste?" 456
```

program treba da ispise:

```
Broj argumenata: 7
0 ./echoargs
1 Ovo
2 je
3 primer.
4 123
5 Kako ste?
6 456
```

(Prikazati varijante programa sa nizovskom i pokazivačkom sintaksom)

3. Datoteka `ulaz.txt` sadrži cele brojeve. Napisati program koji izračunava njihov zbir i upisuje ga u datoteku `zbir.txt`.
4. Napisati program koji sabira cele brojeve navedene u komandnoj liniji (pretpostaviti da su svi navedeni argumenti zaista celi brojevi). Na primer, ako se program pokrene sa

```
./sumargs 123 456 -78 910
```

program treba da ispiše 1411. (Za vežbu dati modifikaciju koja ignoriše one argumente koji nisu ispravno zapisani celi brojevi).

5. Napisati program koji u datoteku čije se ime zadaje kao argument komandne linije upisuje prvih N Fibonačijevih brojeva. Broj N se takođe zadaje kao argument komandne linije. Fibonačijevi brojevi se računaju po formuli

$$F(N+2) = F(N+1) + F(N), \quad F(0)=1, \quad F(1)=1$$

6. Napisati program koji utvrđuje koliko se puta data reč pojavljuje u datoj tekstualnoj datoteci. Naziv datoteke kao i reč zadaju se kao argumenti komandne linije.
7. Datoteka u prvom redu sadrži broj studenata, a u svakom narednom informacije o jednom studentu: ime, prezime, smer, broj ESP bodova. Napisati program koji iz te datoteke, čije se ime navodi kao prvi argument komandne linije, vrši učitavanje liniju po liniju i ispisuje informaciju o studentu koji je sakupio najviše bodova u datoteku sa imenom `kandidatZaStipendiju.txt`.
8. Uobičajena praksa na UNIX sistemima je da se argumenti komandne linije dele na opcije i argumente u užem smislu. Opcije počinju znakom '-' nakon čega obično sledi jedan ili više karaktera koji označavaju koja je opcija u pitanju. Ovim se najčešće upravlja funkcionisanjem programa i neke mogućnosti se uključuju ili isključuju. Argumenti najčešće predstavljaju opisne informacije poput na primer imena datoteka. Napisati program koji ispisuje sve opcije koje su navedene u komandnoj liniji. Na primer, ako se program pokrene sa

```
./echoopts -abc input.txt -d -Fg output
```

program treba da ispiše

```
Prisutne su opcije:
a b c d F g
```

(Prikazati čitljivo rešenje sa nizovskom sintaksom, kao i K&R rešenje sa pokazivačkom sintaksom.)

2.2 Srednje teški zadaci

1. Napisati program koji prepisuje datoteku čije se ime navodi kao prvi argument komandne linije u datoteku čiji se argument navodi kao drugi argument komandne linije programa. Ukoliko je navedena opcija `-u` program prilikom prepisivanja treba da zamenjuje sva mala slova velikim, a ukoliko je navedena opcija `-l` sva velika slova se zamenjuju malim.

(Povezati sa već rešenim zadatkom kod koga su fiksirana imena datoteka. Diskutovati varijantu u kojoj je fiksiran položaj opcije u listi argumenata, kao i varijantu u kojoj se opcija može navesti na proizvoljnom mestu u listi argumenata.)

2. Datoteka `ocene.txt` sadrži imena i prezimena studenata i njihove ocene. Napisati program koji za svakog studenta izračunava i ispisuje prosečnu ocenu. Na primer, ako datoteka sadrži:

```
Pera Peric 8 9 9 8
Mika Mikic 10 6 7 7 8
Ana Anic 10 10 9 10
```

program treba da ispiše:

```
Pera Peric      8.50
Mika Mikic      7.60
Ana Anic        9.75
```

3. Napisati funkciju

```
int gettag(FILE *f, char *s, int size)
```

koja učitava sledeću HTML etiketu iz datoteke koja je predstavljena fajl-pokazivačem `f`, u string `s`, ali ne više od `size` karaktera. Funkcija vraća 1 ako je uspeła da pročita narednu etiketu, ili 0 za neuspeh.

4. Napisati C program koji iz datoteke `dat.txt` čita reč po reč i na standardni izlaz ispisuje reči koje su duže od `N`, ukoliko je prilikom pokretanja programa navedena opcija `-d`, ili reči koje su kraće od `N`, ukoliko je prilikom pokretanja programa navedena opcija `-k`. Broj `N` se zadaje kao argument komandne linije.
5. Napisati C program koji kao argument komandne linije prima ime datoteke u kojoj se nalaze brojevi koji su izvučeni na Lotou, kao i opciju `-r` ili `-o`. Učitati ove brojeve u niz, sortirati ga u rastućem poretku ako je zadata opcija `-r` a u opadajućem ako je zadata `-o`, a zatim ispisati na standardni izlaz. Ukoliko prilikom rada programa dođe do greške, informacije ispisati na izlaz za greške.
6. U datoteci (čije se ime zadaje iz komandne linije) dat je neki tekst. Napisati program koji šifruje dati tekst tzv. Čezarovom šifrom, i rezultujući tekst upisuje u drugu datoteku, čije se ime takođe zadaje iz komandne linije (ako se ne navede, ispisuje na standardni izlaz). Čezarovom šifrom se slova kodiraju na sledeći način:

```
A -> D, B -> E, ..., X -> A, Y -> B, Z -> C (slicno za mala slova).
Ostali karakteri ostaju nepromenjeni.
```

7. U datoteci `studenti.txt` nalaze se za svakog studenta redom prezime i ime. Na postojeća prezimena i imena dodati svoje ime i prezime. Zatim iz ove datoteke učitati studente, sortirati ih u alfabetskom poretku po prezimenu, a u slučaju istog prezimena po imenu. Tako sortirana prezimena i imena upisati u datoteku `sortirano.txt`.
8. U datoteci se nalaze informacije o proizvodima u nekom magacinu. Datoteka je u sledećem formatu:

```
<broj_proizvoda (n)>
<pr_1> <cena_1> <kolicina_1>
<pr_2> <cena_2> <kolicina_2>
...
<pr_n> <cena_n> <kolicina_n>
```

Napisati program koji određuje ukupnu vrednost svih proizvoda u magacinu, kao i pojedinačni proizvod čija je zbirna cena (`cena * količina`) najveća. Ispisati na standardni izlaz proizvode sortirano po zbirnoj ceni. Ime datoteke se zadaje na komandnoj liniji.

3 Kompilacija iz više izvornih datoteka

3.1 Srednje teški zadaci

1. Napisati malu biblioteku za rad sa kompleksnim brojevima:
 - (a) Definirati strukturu `KompleksniBroj` koja predstavlja kompleksni broj reprezentovan svojim realnim i imaginarnim delom.
 - (b) Napisati funkciju koja učitava kompleksni broj sa standardnog ulaza.
 - (c) Napisati funkciju koja ispisuje kompleksni broj na standardni izlaz.
 - (d) Napisati funkciju koja sabira dva kompleksna broja.
 - (e) Napisati funkciju koja oduzima dva kompleksna broja.
 - (f) Napisati funkciju koja množi dva kompleksna broja.
 - (g) Napisati funkciju koja deli dva kompleksna broja.
 - (h) Napisati funkciju koja množi kompleksni broj realnim.
 - (i) Napisati funkciju koja sabira kompleksni i realni broj.
 - (j) Napisati funkciju koja određuje konjugat datog kompleksnog broja.
 - (k) Napisati funkciju koja određuje moduo datog kompleksnog broja.
 - (l) Napisati funkciju koja određuje argument datog kompleksnog broja.

Sve vreme, paralelno sa razvojem funkcija, pisati i glavni program koji ih testira.

Nakon toga, izdvojiti funkcije u zasebnu datoteku `kompleksni.c`, a program u `test-kompleksni.c`. Prikazati probleme u kompilaciji, pošto ne postoji definicija strukture kao ni prototipovi funkcija u `test-kompleksni.c`. Diskutovati da dvostruko navođenje ovoga u obe C datoteke dovodi do velikih problema prilikom održavanja programa. Uvesti `kompleksni.h` kao rešenje i uključiti ga u obe datoteke. Prikazati kreiranje objektnih datoteka `kompleksni.o`. Prikazati:

```
gcc -o test-kompleksni kompleksni.c test-kompleksni.c
```

```
i
```

```
gcc -c kompleksni.c
```

```
gcc -c test-kompleksni.c
```

```
gcc -o test-kompleksni kompleksni.o test-kompleksni.o
```

Pomenuti `make`.

„Distribuirati” datoteke `kompleksni.h` i `kompleksni.o` (ne `kompleksni.c`) - na primer premeštanjem u novi direktorijum. Uz korišćenje ove biblioteke napisati aplikaciju koja rešava kvadratnu jednačinu.

2. Napraviti biblioteku za rad sa skupovima celih brojeva (skupovi imaju najviše 1000 elemenata).
 - (a) Definirati strukturu podataka za reprezentovanje skupova (koristiti sortirane nizove).
 - (b) Napisati funkcije za učitavanje i ispis skupova.
 - (c) Napisati funkciju koja proverava da li skup sadrži dati element.
 - (d) Napisati funkciju koja određuje presek dva skupa.
 - (e) Napisati funkciju koja određuje uniju dva skupa.
 - (f) Napisati funkciju koja određuje razliku dva skupa.

3.2 Teži zadaci

1. Napisati malu biblioteku za rad sa polinomima:
 - (a) Definirati strukturu `Polinom` koja predstavlja polinom (stepena najviše 20). Struktura sadrži stepen i niz koeficijenata. (Diskutovati redosled navođenja koeficijenata u nizu)
 - (b) Napisati funkciju koja ispisuje polinom na standardni izlaz u što lepšem obliku.
 - (c) Napisati funkciju koja učitava polinom sa standardnog ulaza.

- (d) Napisati funkciju za izračunavanje vrednosti polinoma u datoj tački (diskutovati Hornerov algoritam i njegove prednosti).
- (e) Napisati funkciju koja sabira dva polinoma.
- (f) Napisati funkciju koja množi dva polinoma.
- (g) (Naprednijim studentima za vežbu dati zadatak da napišu funkciju koja deli dva polinoma i izračunava NZD dva polinoma.)

Sve vreme, paralelno sa razvojem funkcija, pisati i glavni program koji ih testira.

Nakon toga, izdvojiti funkcije u zasebnu datoteku `polinom.c`, a program u `test-polinom.c`. Prikazati probleme u kompilaciji, pošto ne postoji više definicija strukture kao ni prototipovi funkcija u `test-polinom.c`. Diskutovati da dupliranje ovoga u obe C datoteke dovodi do velikih problema prilikom održavanja programa. Uvesti `polinom.h` kao rešenje i uključiti ga u obe datoteke. Prikazati kreiranje objektna datoteke `polinom.o`.

```
gcc -o test-polinom polinom.c test-polinom.c
```

```
i
```

```
gcc -c polinom.c
```

```
gcc -c test-polinom.c
```

```
gcc -o test-polinom polinom.o test-polinom.o
```

Pomenuti `make`.

„Distribuirati” datoteke `polinom.h` i `polinom.o` (ne `polinom.c`) - na primer premeštanjem u novi direktorijum. Napisati novi test program, prevesti ga i povezati.

2. Napisati malu biblioteku za rad sa velikim prirodnim brojevima (biblioteku razdvojiti u `*.c` i `*.h` datoteku).

- (a) Definirati strukturu `VelikiBroj` kojom se broj reprezentuje nizom cifara (najviše 1000).
- (b) Napisati funkciju za učitavanje velikog broja iz niske.
- (c) Napisati funkciju za ispis velikog broja na standardni izlaz.
- (d) Napisati funkciju za sabiranje dva velika broja.
- (e) Napisati funkciju za poređenje dva velika broja (funkcija vraća -1, 0, ili 1).
- (f) Napisati funkciju za množenje velikog broja cifrom.
- (g) Napisati funkciju za množenje dva velika broja.

Sve vreme, paralelno sa razvojem funkcija, pisati i glavni program koji ih testira. Upotrebiti ovu biblioteku za izračunavanje vrednosti broja 100!.

4 Bitski operatori

4.1 Lakši zadaci

1. Napisati funkciju koja ispisuje sve bitove datog celog broja (diskutovati označeni i neoznačeni slučaj).
2. Napisati funkciju koja broji binarne jedinice u zapisu datog neoznačenog broja.
 - (a) Prikazati varijantu sa pomeranjem maske (linearna složenost).
 - (b) Prikazati varijantu sa pomeranjem broja (linearna složenost).
3. (a) Napisati funkciju koja datom broju uklanja poslednju jedinicu (napomena: razmatrati odnos binarnog zapisa brojeva n i $n - 1$).
- (b) Napisati funkciju koja ispituje da li je dati ceo broj n stepen broja 2.
4. (a) Napisati funkciju koja postavlja poslednjih 8 bitova datog neoznačenog broja na 0.
- (b) Napisati funkciju koja postavlja poslednjih 8 bitova datog neoznačenog broja na 1.
- (c) Napisati funkciju koja izdvaja poslednjih 8 bitova datog neoznačenog celog broja.
- (d) Napisati funkciju koja vraća broj koji se dobija upisivanjem poslednjih 8 bitova broja y na poslednjih 8 bitova broja x .
- (e) Napisati funkciju koja vraća broj koji se dobija invertovanjem poslednjih 8 bitova broja x .
5. Napisati funkciju koja broji koliko se puta kombinacija 11 (dve uzastopne jedinice) pojavljuje u binarnom zapisu celog broja x .

4.2 Srednje teški zadaci

- Napisati funkciju koja određuje najveći broj koji se može zapisati istim binarnim ciframa kao dati broj.
 - Napisati funkciju koja određuje najmanji broj koji se može zapisati istim binarnim ciframa kao dati broj.
- Napisati funkciju koja određuje broj koji se dobija rotiranjem ulevo datog celog broja.
 - Napisati funkciju koja određuje broj koji se dobija rotiranjem udesno datog celog broja (diskutovati neoznačenu i označenu varijantu).
- Napisati funkciju koja određuje broj koji se dobija kada se n bitova datog broja, počevši od pozicije p postave na 0.
 - Napisati funkciju koja određuje broj koji se dobija kada se n bitova datog broja, počevši od pozicije p postave na 1.
 - Napisati funkciju koja izdvaja n bitova datog neoznačenog broja, počevši od pozicije p i vraća ih kao bitove najmanje težine rezultata. Npr. 7 bitova broja 1001011101110110, počevši od pozicije 9 su 1101110, pa rezultat treba da bude 000000001101110.
 - Napisati funkciju koja vraća broj koji se dobija upisivanjem poslednjih n bitova broja y u broj x , počevši od pozicije p .
 - Napisati funkciju koja vraća broj koji se dobija invertovanjem n bitova broja x počevši od pozicije p .
- Napisati funkciju koja određuje broj koji se dobija preslikavanjem datog neoznačenog broja oko svoje sredine (centralna simetrija, slika u ogledalu).
- Napisati program koji ispisuje zapis datog broja u pokretnom zarezu i određuje njegov znak, mantisu i eksponent (napomena: koristiti uniju koja sadrži float i long). Diskutovati IEEE754 standard.
- Napisati funkciju koja na osnovu neoznačenog broja x formira nisku s koji sadrži heksadekadni zapis broja x , koristeći algoritam za brzo prevodjenje binarnog u heksadekadni zapis (svake 4 binarne cifre se zamenjuju jednom odgovarajućom heksadekadnom cifrom).
- Napisati funkciju koja za dva data broja x i y invertuje u podatku x one bitove koji se poklapaju sa odgovarajućim bitovima u broju y . Ostali bitovi ostaju nepromenjeni.
- Napisati funkciju koja očitava i na ekranu prikazuje binarni zapis float podatka x . Upustvo: koristiti uniju.
- Napisati funkciju koja za podatak proizvoljnog tipa prikazuje njegov binarni zapis na ekranu. Uputstvo: koristiti void pokazivač i konverzije pokazivača.

4.3 Teži zadaci

- Napisati funkciju koja broji binarne jedinice u zapisu datog neoznačenog broja.
 - Prikazati varijantu u kojoj se u svakom koraku uklanja poslednja jedinica. (linearna složenost - broj koraka jednak broju jedinica)
 - Prikazati paralelnu varijantu za 32 bita (logaritamska složenost). Napomena: sabirati 16 parova jednobitnih brojeva i dobiti 16 dvobitnih brojeva, zatim sabirati 8 parova dvobitnih brojeva i dobiti 8 četvorobitnih brojeva, zatim sabirati 4 para četvorobitnih brojeva i dobiti četiri osmobitna broja, zatim sabirati dva para osmobitnih brojeva i dobiti dva šesnaestobitna broja čijim se sabiranjem dobija jedan tridesetdvobitni broj koji predstavlja konačnu traženu sumu jedinica. Npr.

```
01011011001010100110110101011011
010101100001010101100101010110
00100011000100100010001100100011
00000101000000110000010100000101
0000000000001000000000000001010
000000000000000000000000010010
```

- Napisati funkciju koja određuje ceo deo logaritma za osnovu 2 datog neoznačenog celog broja.
 - Prikazati varijantu sa pomeranjem broja udesno dok ne postane 0 (linearna složenost).
 - Prikazati varijantu sa binarnom pretragom (logaritamska složenost).

5 Sortiranje i pretraga

5.1 Lakši zadaci

1. Jedan birački spisak je sortiran po imenu građana, a drugi po prezimenu građana. Napisati program koji učitava spisak građana (najviše 1000) i ispisuje koliko građana ima isti redni broj u oba biračka spiska.
2. U datom nizu brojeva pronaći dva broja na najmanjem rastojanju (uputstvo: prvo sortirati niz).
3. Napisati funkciju koja određuje nulu funkcije $\cos(x)$ na intervalu $[0, 2]$ metodom polovljenja intervala (napomena: analogija sa binarnom pretragom).
4. Napisati program koji pronalazi broj koji se najveći broj puta pojavljuje u nizu (uputstvo: prvo sortirati niz, a zatim naći najdužu sekvencu jednakih elemenata).
5. Napisati funkciju koja sortira niz niski (najviše 1000):
 - (a) leksikografski
 - (b) po dužini
 - (c) po dužini, pri čemu se reči iste dužine sortiraju leksikografski.

(Napomena: koristiti bilo koji algoritam sortiranja, a za domaći zahtevati da studenti izmene program implementirajući neki drugi algoritam sortiranja. Izdvojiti poređenja u zasebne funkciju. Napisati i program koji testira napravljene funkcije.)

5.2 Srednje teški zadaci

1. Iz datoteke `artikli.txt` učitava se niz podataka o artiklima jedne prodavnice (najviše 100000). Za svaki artikl, poznat je jedinstveni (bar) kôd, naziv, naziv proizvođača i cena. Napisati program koji simulira rad kase u prodavnici. Kasir unosi kodove jednog po jednog artikla, a program ispisuje detaljnije podatke o artiklima. Nakon unosa koda 0, završava se tekući račun i ispisuje ukupna cena svih unetih artikala. Podatke o artiklima čuvati u okviru niza struktura uređenog rastuće po kodovima i za pronalaženje detalja o artiklu koristiti binarnu pretragu.
2. Napisati program koji uklanja duplikate iz datog niza e-mail adresa (uputstvo: prvo sortirati niz).
3. Napisati program koji upoređuje efikasnost linearne i binarne pretrage. Kao argument komandne linije se navode brojevi $n \leq 10^6$ i broj x . Program kreira sortiran niz slučajnih celih brojeva dužine n i proverava da li se da li niz sadrži broj x . Program ispisuje vreme potrebno funkciji koja implementira linearnu pretragu i funkciji koja implementira binarnu pretragu. Vreme meriti korišćenjem funkcija iz `time.h`. Analizirati porast vremena sa porastom n .
4. Napraviti biblioteku „sort.h” i „sort.c” koja implementira algoritme sortiranja nizova celih brojeva. Biblioteka treba da sadrži bubble, selection, insertion i shell sort. Upotrebiti biblioteku kako bi se napravilo poređenje efikasnosti različitih algoritama sortiranja. Efikasnost meriti na slučajno generisanim nizovima, na već sortiranim nizovima i na naopako sortiranim nizovima. Izbor algoritma, veličine i početnog rasporeda elemenata niza birati kroz argumente komandne linije. Vreme meriti programom `time`. Analizirati porast vremena sa porastom dimenzije n . Porediti vremena za različite implementacije istog algoritma.
5. Napisati funkciju koja ispituje da li u datom nizu postoje dva elementa čiji je zbir jednak datom broju (uputstvo: prvo sortirati niz).
6.
 - (a) Napisati program koji ispituje da li su dve unete niske anagrami (uputstvo: normalizovati sortiranjem).
 - (b) Napisati program koji pronalazi sve anagrame u datom nizu reči
7. Napisati funkciju koja prima dva sortirana niza, i na osnovu njih pravi novi sortirani niz koji koji sadrži elemente oba niza

```
int merge(int *niz1, int dim1, int *niz2, int dim2, int *niz3, int dim3);
```

Treća dimenzija predstavlja veličinu niza u kog se smešta rezultat. Ako je ona manja od potrebne dužine, funkcija vraća -1, kao indikator neuspeha, inače vraća 0.

5.3 Teži zadaci

1. Razmatrajmo dve operacije: operacija U je unos novog broja x , a operacija N određivanje n -tog po veličini od unetih brojeva. Implementirati program koji izvršava ove operacije (Može postojati najviše 100000 operacija unosa, a uneti elementi se mogu ponavljati, pri čemu se i ponavljanja računaju prilikom brojanja). Na primer:

```
Ulaz: U 2 U 0 U 6 U 4 N 1 U 8 N 2 N 5 U 2 N 3 N 5
Izlaz: 0 2 8 2 6
```

(Napomena: brojeve čuvati u sortiranom nizu i svaki naredni element umetati na svoje mesto)

Optimizovati program, ukoliko se zna da neće biti više od 500 različitih unetih brojeva.

2. Sa dve susedne stranice pravougaone livade dve grupe krtica istovremeno kreću da kopaju tunele (jedna grupa na gore, a druga na desno). Krtica prestaje da kopa ukoliko naiđe na već iskopan tunel (npr. krticu A zaustavlja krtica C, krticu C i D zaustavlja krtica B, a krticu B zaustavlja krtica E). Za svaku krticu se zna udaljenost od ćoška livade i brzina kojom kopa.

```
|EEEEEEEEEE
|      B
|DDDDDDDB
|      B
|CCCCCCB
|   A   B
|_A_B_
```

Napisati program koji određuje koliko dugo svaka krtica kopa. Unose se broj krtica u obe grupe, a zatim udaljenost i brzina za svaku krticu. Izlaz je vreme za svaku krticu prikazano na dve decimale (-1 ukoliko se ne zaustavlja) u istom redosledu u kojem su krtice unete. Npr.

```
Ulaz:
2 3
1 4
2 5.1
4.5 3.6
1 1
5 0.5
```

```
Izlaz:
1.12
0.88
-1.00
1.00
-1.00
```

3. Šef u restoranu je neuredan i palačinke koje ispeče ne slaže redom po veličini. Konobar pre serviranja mora da sortira palačinke po veličini, a jedina operacija koju sme da izvodi je da obrne deo palačinki. Na primer:

```
3   5   2   1
4   4   1__ 2
5__ 3   3   3
1   1   4   4
2   2__ 5   5
```

Napisati program koji u najviše $2n - 3$ okretanja sortira učitani niz. (Uputstvo: imitirati selection sort i u svakom koraku dovesti jednu palačinku na svoje mesto korišćenjem najviše dva okretanja.)

4. U prvom kvadrantu dato je $1 \leq N \leq 10000$ duži svojim koordinatama (duži mogu da se seku, preklapaju, itd.). Napisati program koji pronalazi najmanji ugao $0 \leq \alpha \leq 90^\circ$, na dve decimale, takav da je suma dužina duži sa obe strane polupoluprave iz koordinatnog početka pod uglom α jednak (neke duži bivaju presečene, a neke ne). Npr.

```

Ulaz:
2
2 0 2 1
1 2 2 2

Izlaz:
26.57

```

(Uputstvo: vršiti binarnu pretragu intervala $[0, 90^\circ]$).

6 Rekurzija

6.1 Lakši zadaci

1. (a) Napisati rekurzivnu funkciju koja štampa brojeve između 0 i n .
(b) Napisati rekurzivnu funkciju koja štampa brojeve između n i 0.
2. Napisati rekurzivnu funkciju koja izračunava sumu prvih n prirodnih brojeva.
3. Napisati rekurzivnu funkciju koja izračunava x^k , za dati realni broj x i prirodan broj k . (Diskutovati rešenja linearne i logaritamske složenosti)
4. Napisati rekurzivnu funkciju koja sabira niz celih brojeva.
5. Napisati rekurzivnu funkciju koja određuje maksimum niza celih brojeva.
6. Napisati rekurzivnu funkciju koja izračunava skalarni proizvod dva data vektora.
7. Napisati rekurzivnu funkciju koja sabira dekadne cifre datog celog broja x .
8. Napisati repno-rekurzivnu funkciju koja izračunava $n!$.
9. Napisati rekurzivnu funkciju koja računa broj pojavljivanja elementa x u nizu a dužine n .
10. Napisati rekurzivnu funkciju kojom se proverava da li je broj n prost broj.
11. Napisati rekurzivnu funkciju kojom se računa suma delilaca datog broja.

6.2 Srednje teški zadaci

1. Napisati rekurzivnu funkciju za određivanje NZD dva broja Euklidovim algoritmom.
2. Napisati rekurzivnu funkciju koja izračunava vrednost binomnog koeficienta $\binom{n}{k}$.
3. (a) Napisati rekurzivnu funkciju koja prikazuje dekadne cifre datog celog broja.
(b) Napisati rekurzivnu funkciju koja prikazuje dekadne cifre datog celog broja u obrnutom poretaku.
4. Napisati rekurzivnu funkciju koja unosi reč (karakter po karakter do prvog razmaka) i štampa je obrnuto (ne koristiti nizove).
5. Napisati rekurzivnu funkciju koja ispituje da li je data niska palindrom.
6. Napisati rekurzivnu funkciju koja može da izračuna 2010.-ti član Fibonačijevog niza ($f_0 = 1$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$, $n \geq 2$). (Napomena: koristiti biblioteku za rad sa velikim brojevima. Prikazati dinamičko rešenje, tj. memoizaciju. Prikazati funkciju koja kroz repne rekurzivne pozive prosleđuje dva prethodna broja niza.)
7. Paskal-ov trougao se dobija tako što mu je svako polje (izuzev 0-te vrste i 0-kolone zbir jednog polja levo i jednog polja iznad.

```

1 1 1 1 1 1
1 2 3 4 5
1 3 6 10
1 4 10
1 5
1

```

- (a) Napisati rekursivnu funkciju koja izračunava d_n kao sumu elemenata n -te hipotenuze pravouglog trougla.
- (b) Napisati rekursivnu funkciju koja izračunava vrednost polja (i, j) .
8. Fibonačijev trougao se dobija tako što mu je svako polje (izuzev 0-te vrste i 0-kolone zbir jednog polja levo i jednog polja iznad.

```

1  1  1  1  1  1
1  2  3  4  5
2  5  9 14
3 10 22
5 20
8

```

- (a) Napisati rekursivnu funkciju koja izračunava d_n kao sumu elemenata n -te hipotenuze pravouglog trougla.
- (b) Napisati rekursivnu funkciju koja izračunava vrednost polja (i, j) .
9. Napisati rekursivnu funkciju za uzračunavanje determinante reda n :

$$D_n = \begin{vmatrix} a & b & 0 & 0 & \dots & 0 & 0 \\ c & a & b & 0 & \dots & 0 & 0 \\ 0 & c & a & b & \dots & 0 & 0 \\ 0 & 0 & c & a & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a & b \\ 0 & 0 & 0 & 0 & \dots & c & a \end{vmatrix}$$

10. Napisati rekursivnu funkciju koja računa sumu elemenata na parnim pozicijama u nizu celih brojeva. Funkcija treba da bude posredno rekursivna, tj. rekursija se ostvaruje preko posredno rekursivne funkcije koja računa sumu elemenata na neparnim pozicijama.
11. Napisati rekursivnu funkciju koja računa n -ti element u Fibonačijevom nizu. Popraviti funkciju tako da se problemi manje dimenzije rešavaju samo jedanput. (dinamičko programiranje)
12. Napisati rekursivnu funkciju koja za dati ceo broj n i dužinu d određuje podbrojeve broja n dužine d . Npr. za broj $n=23456$ i $d=2$ rezultat treba da bude 56 45 34 23.
13. Napisati rekursivnu funkciju kojom se proverava da li su tri zadata broja uzastopni članovi niza.

6.3 Teži zadaci

- Korišćenjem identiteta $\sqrt{4 \cdot x} = 2 \cdot \sqrt{x}$ napisati rekursivnu funkciju koja izračunava ceo deo korena datog broja.
- Napisati rekursivnu funkciju koja prikazuje sve permutacije skupa $\{1, 2, \dots, n\}$.
- Napisati rekursivnu funkciju koja prikazuje sve varijacije sa ponavljanjem dužine n skupa $\{0, 1\}$.
- Problem "Hanojskih kula": Data su tri vertikalna štapa, na jednom se nalazi n diskova poluprečnika 1,2,3,... do n , tako da se najveći nalazi na dnu, a najmanji na vrhu. Ostala dva štapa su prazna. Potrebno je premestiti diskove na drugi štap tako da budu u istom redosledu, premestajući jedan po jedan disk, pri čemu se ni u jednom trenutku ne sme staviti veći disk preko manjeg.

7 Rekursivne implementacije algoritama pretrage i sortiranja

7.1 Lakši zadaci

- Napisati rekursivne implementacije funkcija za linearnu i binarnu pretragu.
- Napisati funkciju koja dati niz brojeva sortira tako da prvo idu negativni brojevi (u proizvoljnom redosledu), zatim nule i na kraju negativni brojevi.

7.2 Srednje teški zadaci

1. Napisati rekurzivne implementacije funkcija za sortiranje (selection, insertion) niza celih brojeva
2. Napisati rekurzivnu funkciju koja „meša” (eng. shuffle) dati niz brojeva.
3. Napisati rekurzivnu implementaciju QuickSort algoritma.
4. Napisati rekurzivnu implementaciju MergeSort algoritma.

8 Pokazivači na funkcije

8.1 Lakši zadaci

1. Napisati funkciju koja „tabelira” datu realnu funkciju realnog argumenta $f(x)$, tj. izračunava vrednosti date funkcije na diskretnoj ekvidistantnoj mreži od n tačaka intervala $[a, b]$.
2. Napisati funkciju koja izračunava parcijalnu sumu $\sum_{i=0}^n a_i$, pri čemu se zadaju n i a_i (u obliku funkcije).
3. Napisati funkciju koja metodom polovljenja intervala približno prolazi nulu date funkcije $f(x)$ na intervalu $[a, b]$. Pretpostavlja se da je f neprekidna i da je $f(a) \cdot f(b) < 0$. (Naglasiti vezu sa binarnom pretragom)

8.2 Srednje teški zadaci

1. Napisati funkciju koja približno izračunava integral funkcije $f(x)$ na intervalu $[a, b]$. Funkcija f se prosleđuje kao parametar, a integral se procenjuje po Simpsonovoj formuli:

$$I = \frac{h}{3} \left(f(a) + 4 \sum_{i=1}^n f(a + (2i-1)h) + 2 \sum_{i=1}^n f(a + 2ih) + f(b) \right)$$

2. Napisati funkciju koja metodom sečice približno pronalazi nulu date funkcije $f(x)$ na intervalu $[a, b]$. Pretpostavlja se da je f neprekidna i da je $f(a) \cdot f(b) < 0$. (Naglasiti vezu sa interpolacionom pretragom)
3. Napisati generičku funkciju koja sortira niz elemenata proizvoljnog tipa. Funkcija prima adresu početka niza, broj članova, veličinu jednog člana niza i funkciju koja poredi dva elementa niza.

9 Dinamička alokacija memorije

9.1 Lakši zadaci

1. Napisati program koji učitava brojeve sa standardnog ulaza u niz, a zatim ispisuje niz unazad. Pre učitavanja brojeva, učitava se njihov broj, pri čemu taj broj nije unapred poznati ni ograničen.
2. Napisati funkciju `char* concatenate(char* s, char* t)` koja nadovezuje dve date niske.

9.2 Srednje teški zadaci

1. Napisati program koji učitava brojeve sa standardnog ulaza u niz, a zatim ispisuje niz unazad. Brojevi se unose sve dok se ne unese 0, pri čemu njihov broj nije unapred poznat ni ograničen. (Napomena: vršiti realokaciju niza dok se elementi unose. Pokazati varijantu sa ručnom realokacijom — kopiranjem i brisanjem, i sa korišćenjem funkcije `realloc` i diskutovati prednosti.)
2. U datoteci *pesme.txt* nalaze se informacije o gledanosti pesama na Youtube-u u sledećem formatu:

izvodjac - naslov, brojGledanja

Napisati program koji učitava informacije o pesmama i vrši sortiranje pesama u zavisnosti od argumenata komandne linije:

- (a) nema opcija, sortiranje po broju gledanja.
- (b) `-i`, sortiranje po imenu izvođača
- (c) `-n`, sortiranje po naslovu pesme

Ne praviti nikakve pretpostavke o maksimalnoj dužini imena izvođača, naslova pesme, kao i ukupnom broju pesama.

3. Pera sakuplja sličice igrača za predstojeće Svetsko prvenstvo u fudbalu. U datoteci *slicice.txt* se nalaze informacije o sličicama koje mu nedostaju u formatu:

redni_broj_slicice ime_reprezentacije_kojoj_slicica_pripada

Pomozite Peri da otkrije koliko mu sličica ukupno nedostaje, kao i da pronadje ime reprezentacije čijih sličica ima najmanje.

Zadatak realizovati korišćenjem `realloc` funkcije.

4. U datoteci *temena.txt* se nalaze tačke koje predstavljaju temena nekog n -tougla. Napisati program koji na osnovu sadržaja datoteke ispisuje o kom n -touglu je reč, a zatim i vrednosti njegovog obima i površine. Npr. Za datoteku *temena.txt* sa sadržajem

1.0 0.0

0.0 0.0

0.0 1.0

ispis je:

n: 3

obim: 3.41

povrsina: 0.5

10 Višedimenzioni nizovi i matrice

10.1 Lakši zadaci

1. Napisati program koji učitava realnu matricu, a zatim izračunava njen trag (sumu elemenata na dijagonali), gornju vandijagonalnu normu (sumu apsolutnih vrednosti elemenata iznad dijagonale) i Euklidsku normu (koren iz sume kvadrata svih njenih elemenata).

(a) Matrica je dimenzije 10×10 .

(b) Matrica je dimenzije $n \times n$, pri čemu se n unosi pre samog unosa matrice i nije unapred ograničeno.

2. Relacije su predstavljene kvadratnim matricama nula ili jedinica.

(a) Napisati funkciju koja proverava da li je data relacija refleksivna.

(b) Napisati funkciju koja proverava da li je data relacija simetrična.

(c) Napisati funkciju koja proverava da li je data relacija tranzitivna.

(d) Napisati funkciju koja određuje refleksivno zatvorenje relacije (najmanju refleksivnu relaciju koja sadrži datu).

(e) Napisati funkciju koja određuje simetrično zatvorenje relacije (najmanju simetričnu relaciju koja sadrži datu).

(f) Napisati funkciju koja određuje refleksivno-tranzitivno zatvorenje relacije (najmanju refleksivnu i tranzitivnu relaciju koja sadrži datu). (Napomena: efikasnost nije presudna - ukazati na Varšalov algoritam).

Diskutovati statičku i dinamičku alokaciju.

3. Napisati program koji sa ulaza učitava matricu dimenzija $m \times n$ i na standardnom izlazu ispisuje:

(a) Vrednost najvećeg elementa na sporednoj dijagonali.

(b) Indeks kolone koja sadrži najmanji element matrice.

(c) Indeks vrste koja sadrži najveći element matrice.

(d) Broj negativnih elemenata matrice.

Dimenzije m i n se zadaju kao argumenti komandne linije.

4. Na ulazu se zadaje ceo broj n , a zatim i dve kvadratne matrice dimenzije n .

(a) Napisati funkciju kojom se proverava da li su zadate matrice jednake.

- (b) Napisati funkciju kojom se izračunava zbir unetih matrica.
- (c) Napisati funkciju kojom se izračunava proizvod unetih matrica.

Napisati program koji testira rad napisanih funkcija. (Prilikom testiranja zgodno je koristiti preusmeravanje ulaza.)

5. Napisati funkciju kojom se proverava da li je zadata matrica A dimenzije n ortonormirana. Matrica je ortonormirana ako je skalarni proizvod svakog para različitih vrsta jednak 0, a skalarni proizvod vrste sa samom sobom 1.

10.2 Srednje teški zadaci

1. Napisati funkciju koja izračunava k -ti stepen kvadratne matrice dimenzije $n \times n$. Voditi računa da se prilikom stepenovanja izvrši što manji broj množenja.
2. Napisati program koji spiralno ispisuje elemente matrice. Npr. matrica

```
1  2  3  4  5
6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
```

se ispisuje kao:

```
1 2 3 4 5 10 15 20 25 24 23 22 21 16 11 6
7 8 9 14 19 18 17 12
13
```

3. Napisati program koji sortira vrste matrice, rastuće na osnovu sume elemenata u vrsti.
4. Napisati program koji na osnovu dveju matrica dimenzija $m \times n$ formira matricu dimenzija $2m \times n$ tako što naizmenično kombinuje jednu vrstu prve matrice i jednu vrstu druge matrice. Matrice su zapisane u datoteci *matrice.txt*. U prvom redu se nalaze dimenzije m i n , u narednih n redova su vrste prve matrice, a u narednih n redova vrste druge matrice. Npr.
Za datoteku *matrice.txt* sa sadržajem

```
2 4
1 2 3 4
8 8 1 2
5 6 3 2
9 0 1 6
```

treba ispisati matricu

```
1 2 3 4
5 6 3 2
8 8 1 2
9 0 1 6
```

5. Napisati program koji sortira kolone matrice, opadajuće, na osnovu vrednosti prvog elementa kolone.
6. Napisati program koji izračunava determinantu matrice dimenzije n , korišćenjem Laplasovog razvoja (rekurzivno). Matrice potrebnih dimenzija alocirati dinamički. (Naglasiti da ovo nije najefikasnija procedura)
7. Na ulazu se zadaje niz celih brojeva koji se završava nulom. Napisati program koji od zadatog niz formira matricu tako da prva vrsta odgovara unetom nizu, a svaka naredna se dobija cikličkim pomeranjem elemenata niza za jednu poziciju ulevo. Npr.
za ulaz
1 2 3 4 0
treba formirati matricu

```
1 2 3 4
2 3 4 1
3 4 1 2
4 1 2 3
```

8. Napisati program koji (bilo kojim algoritmom sortiranja) sortira linije ulazne datoteke.
 - (a) Zna se da ima najviše 1000 linija, a da svaka linija ima najviše 80 karaktera. Za smeštanje sadržaja datoteke koristiti statički alociranu matricu karaktera.
 - (b) Zna se da ima najviše 1000 linija, a da svaka linija ima najviše 80 karaktera. Statički alocirati niz pokazivača na linije, a linije dinamički alocirati (npr. korišćenjem `strdup`) tokom čitanja sadržaja datoteke.
 - (c) Ne zna se ni maksimalna dužina jedne linije, ni broj linija.

11 Funkcije `lfind`, `lsearch`, `bsearch` i `qsort`

11.1 Lakši zadaci

1. Napisati program koji učitava niz linija iz datoteke (najviše 1000 linija sa najviše 80 karaktera), eliminišući duplikate. Koristiti funkciju `lsearch`.
2. Korišćenjem funkcije `qsort` napisati program koji sortira:
 - (a) Niz brojeva opadajuć.
 - (b) Niz prirodnih brojeva po broju delilaca.
 - (c) Niz niski leksikografski.
 - (d) Niz niski opadajuće leksikografski.
 - (e) Niz niski po dužini.
 - (f) Niz niski opadajuće po broju samoglasnika.
 - (g) Niz niski koje sadrže datume u obliku `dd/mm/gg` na osnovu datuma.
 - (h) Niz tačaka na osnovu njihovog rastojanja od kordinatnog početka.
 - (i) Niz tačaka na osnovu ugla koji zaklapaju sa x osom.

Proširiti program tako da na tako sortiran niz, primeni binarnu pretragu, koristeći funkciju `bsearch`.

11.2 Srednje teški zadaci

1. Datoteka `studenti.txt` sadrži spisak studenata. Za svakog studenta poznat je nalog na Alas-u (oblika npr. `mr97125`, `mm09001`), ime i prezime i broj poena. Napisati program koji sortira (korišćenjem funkcije `qsort`) studente po broju poena (ukoliko je prisutna opcija `-p`) ili po nalogu (ukoliko je prisutna opcija `-n`). Studenti se po nalogu sortiraju tako što se sortiraju na osnovu godine, zatim na osnovu smera, i na kraju na osnovu broja indeksa. Ukoliko je u komandnoj liniji uz opciju `-n` naveden i neki broj indeksa, funkcijom `bsearch` potražiti i prijaviti broj poena studenta sa tim nalogom.

12 Liste.

12.1 Lakši zadaci

1. Napisati program koji učitava brojeve sve dok se ne unese 0, smešta elemente u jednostruko povezanu listu i ispisuje ih unazad. (Napomena: ceo zadatak prvo uraditi u okviru `main`, a zatim izdvojiti funkciju za umetanje na početak liste.)
2. Napisati rekurzivnu i nerekurzivnu funkciju koja ispisuje listu celih brojeva.
3. Napisati rekurzivnu i nerekurzivnu funkciju koja briše listu celih brojeva.
4. Napisati rekurzivnu i nerekurzivnu funkciju koja ubacuje element na kraj liste celih brojeva.
5. Napisati rekurzivnu i nerekurzivnu funkciju koja ubacuje element na svoje mesto u sortiranoj listi celih brojeva.
6. Napisati funkciju koja obrće datu jednostruko povezanu listu. Funkcija ne treba da kreira nove čvorove, već da samo postojeće čvorove obrnuto rasporedi.
7. Napisati program koji objedinjuje dve sortirane liste. Funkcija ne treba da kreira nove čvorove, već da samo postojeće čvorove preraspodeli.

8. Neka su date dve jednostruko povezane liste L_1 i L_2 . Napisati funkciju koja od tih lista formira novu listu L koja sadrži alternirajući raspoređene elemente lista L_1 i L_2 (prvi element iz L_1 , prvi element iz L_2 , drugi element L_1 , drugi element L_2 , itd.). Ne formirati nove čvorove, već samo postojeće čvorove rasporediti u jednu listu.
9. Napisati program koji učitava brojeve sve dok se ne unese 0, smešta elemente u dvostruko povezanu listu i ispisuje ih unazad.

12.2 Srednje teški zadaci

1. Napisati funkciju koja formira listu studenata tako što se prvo unosi šifra i ukoliko je ona različita od 0, onda se unose ime i prezime, a u suprotnom se prekida formiranje liste. Za svakog studenta pamtit i ime, prezime i šifru. Ispisati zatim tako formiranu listu i potom osloboditi memoriju.

```
typedef struct _Element
{
    unsigned sifra;
    char ime[20];
    char prezime[20];
    struct _Element *sledeci;
} Element;
```

2. Napisati funkciju koja formira listu studenata tako što se prvo unosi šifra i ukoliko je ona različita od 0, onda se unose ime i prezime, a u suprotnom se prekida formiranje liste. Za svakog studenta pamtit i ime, prezime i šifru. Ispisati zatim tako formiranu listu i potom osloboditi memoriju (dužina imena i prezimena studenta nije unapred ograničena, tako da ih treba alocirati dinamički, a kasnije osloboditi).

```
typedef struct _Element
{
    unsigned sifra;
    char *ime;
    char *prezime;
    struct _Element *sledeci;
} Element;
```

3. Grupa od n plesača (na čijim kostimima su u smeru kazaljke na satu redom brojevi od 1 do n) izvodi svoju plesnu tačku tako što formiraju krug iz kog najpre izlazi k -ti plesač (odbrojava se počev od plesača označenog brojem 1 u smeru kretanja kazaljke na satu). Preostali plesači obrazuju manji krug iz kog opet izlazi k -ti plesač (odbrojava se počev od sledećeg suseda prethodno izbačenog, opet u smeru kazaljke na satu). Izlasci iz kruga se nastavljaju sve dok svi plesači ne budu isključeni. Celi brojevi n , k ($k < n$) se učitavaju sa standardnog ulaza. Napisati program koji će na standardni izlaz ispisati redne brojeve plesača u redosledu napuštanja kruga. PRIMER: za $n = 5$, $k = 3$ redosled izlaska je 3 1 5 2 4.
4. Napisati program koji prebrojava pojavljivanja za svaku od reči datoteke čije se ime zadaje kao argument komandne linije. Zadatak rešiti korišćenjem uredene liste: napisati funkciju koja pretražuje listu i ukoliko pronade reč u listi povećava njen broj pojavljivanja a u slučaju da se reč nije ranije pojavljivala dodaje tu reč na odgo- varajuće mesto u listi tako da lista ostane leksikografski sortirana. Sadržaj rezultujuće liste ispisati na standardni izlaz. Za formiranje liste koristiti strukturu:

```
typedef struct _Element
{
    unsigned broj_pojavljivanja;
    char rec[20];
    struct _Element *sledeci;
} Element;
```

5. Napisati program koji prebrojava pojavljivanja etiketa (bez atributa) HTML datoteke čije se ime zadaje kao argument komandne linije. Sadržaj rezultujuće liste ispisati na standardni izlaz.
6. Neka je dat pokazivač na početak dvostruko povezane liste sortirane u rastućem redosledu koja sadrži pozitivne cele brojeve. Napisati funkciju koja između svih ovih elemenata liste koji se po vrednosti razlikuju za više od 1 ubacuje u datu listu nove elemente tako da lista posle poziva funkcije ima u sebi sukcesivno cele brojeve. Na primer, ako lista sadrži 3, 5, 8 nakon poziva ove funkcije treba da sadrži 3, 4, 5, 6, 7, 8. Napisati zatim program koji učitava brojeve, ne obavezno sortirane, sve dok se ne unese 0, smešta elemente u dvostruko povezanu listu, tako da budu sortirani u rastućem redosledu, poziva prethodnu funkciju i ispisuje elemente nove liste u opadajućem redosledu.

13 Binarna stabla.

13.1 Lakši zadaci

1. Napisati sledeće funkcije za rad sa binarnim stablima koja sadrže cele brojeve (ne obavezno pretraživačkim):
 - (a) Funkcija koja izračunava broj čvorova stabla.
 - (b) Funkcija koja izračunava broj listova stabla.
 - (c) Funkcija koja izračunava sumu čvorova stabla.
 - (d) Funkcija koja izračunava dubinu stabla.
 - (e) Funkcija koja izračunava najveći element u stablu.
 - (f) Funkcija koja ispisuje sve elemente na i -tom nivou.
2. Napisati sledeće funkcije za rad sa binarnim stablima koja sadrže cele brojeve (ne obavezno pretraživačkim):
 - (a) Funkcija koja vraća pokazivač na čvor u stablu kod koga je najmanji (najveći) proizvod sadržaja čvorova iz njegovog desnog podstabla.
 - (b) Funkciju koja vraća pokazivač na čvor u stablu kod koga je najmanja suma sadržaja njegovog levog podstabla.
 - (c) Funkciju koja štampa sadržaj svih čvorova binarnog stabla na putanji od korena do najdubljeg čvora.
 - (d) Funkciju koja štampa sadržaj svih čvorova binarnog stabla na putanji od korena do čvora koji ima najmanju vrednost u stablu.
3. Napisati funkciju koja računa sumu svih vrednosti y u binarnom stablu pretrage koje su manje ili jednake od date vrednosti x . Napisati potom i program koji testira ovu funkciju.
4. Neka je dato binarno stablo koje predstavlja neki izraz (listovi su operandi, a unutrašnji čvorovi su binarni operatori). Napisati funkcije koje prihvataju pokazivač na koreni čvor i ispisuju prefiksni, infiksni i postfiksni prolaz.

Prefiksni prolaz: ispisuje se koren, levo podstablo, pa desno podstablo.

Infiksni prolaz: ispisuje se levo podstablo, koren, pa desno podstablo.

Postfiksni prolaz: ispisuje se levo podstablo, desno podstablo, pa koren.
5. Dva binarna stabla su identična ako su ista po strukturi i sadržaju, odnosno oba korena imaju isti sadržaj i njihova odgovarajuća podstabla su identična. Napistati funkciju koja proverava da li su dva binarna stabla identična.
6. Napisati program koji implementira telefonski imenik. U fajlu, koji se da zadaje kao argument komandne linije, nalazi se niz linija oblika:

`Ime Prezime ***/**-****`

tj. za svaku osobu je naveden broj telefona (pretpostaviti da je datoteka ispravna). Kada se program pokrene, korisnik treba da unese ime i prezime, a program treba da pronade broj telefona te osobe, ili da obavesti korisnika da se osoba ne nalazi u imeniku. Ovaj postupak treba ponavljati dokle god korisnik ne unese komandu za kraj rada. Informacije o brojevima telefona uneti u mapu koja je implementirana preko binarnog stabla pretrage.
7. Sortirati niz celih brojeva koristeći binarno pretraživačko stablo (vremenska složenost $O(n \log n)$).

13.2 Srednje teški zadaci

1. Napisati program koji broji pojavljivanja različitih etiketa u HTML datoteci (koristiti binarno pretraživačko stablo). Dopuniti program tako da ispisuje n etiketa koje se najčešće pojavljuju, opadajući po broju pojavljivanja.
2. Napisati biblioteku koja implementira rad sa skupovima celih brojeva korišćenjem binarnih stabala. Obezbediti funkciju konstrukcije skupa na osnovu datog niza, provere da li element pripada skupu, proveru da li je jedan skup podskup drugog, funkciju za dodavanje elementa skupu, funkcije za pronalaženje unije, preseka i razlike dva skupa, funkciju za ispis skupa, itd.
3. Dva binarna stabla su “slična kao u ogledalu” ako su oba prazna ili ako nisu prazna, ako je levo stablo svakog stabla “slično kao u ogledalu” desnom stablu onog drugog. Napisati funkciju koja proverava da li su dva binarna stabla “slična kao u ogledalu”.

4. Neka je dat pokazivač na koren binarnog stabla čiji čvorovi sadrže cele brojeve. Napisati sledeće funkcije:
 - (a) Funkciju koja vraća broj čvorova koji su po sadržaju veći od svih svojih potomaka.
 - (b) Funkciju koja ispisuje čvorove koji su veći od sume svih svojih potomaka.
5. Binarno stablo se naziva HEAP ako za svaki čvor u stablu važi da je njegov sadržaj veći od sadržaja svih ostalih čvorova u njegovim podstablama. Napisati funkciju koja proverava da li je dato binarno stablo celih brojeva HEAP.
6. AVL-stablo je binarno stablo pretrage kod koga apsolutna razlika visina levog i desnog podstabla svakog elementa nije veća od jedan. Napisati funkciju koja kao argument uzima pokazivač na koren nekog binarnog stabla i vraća broj čvorova koji ispunjavaju uslove za AVL stablo.
7. Napisati program koji implementira podsetnik za rođendane. U fajlu se nalazi niz linija oblika:

Ime Prezime DD. MM. YYYY.

tj. za svaku osobu je naveden datum rođenja. Kada se program pokrene, korisnik treba da unese datum, a program treba da pronađe osobu čiji je rođendan najbliži zadatom datumu. Ovaj postupak treba ponavljati dokle god korisnik ne unese komandu za kraj rada. Informacije o rođendanima uneti u mapu koja je implementirana preko binarnog stabla pretrage.

14 Stekovi, redovi.

14.1 Lakši zadaci

1. Napisati funkcije koje implementiraju operacije “stavi na stek” i “skini sa vrha steka” i funkcije koje ispituju stanje steka “prazan” i “pun”, nad stekom koji je implementiran kao niz.
2. Napisati funkciju koja računa vrednost postfiksno izraza nad pozitivnim celim brojevima pomoću steka. Dozvoljene operacije su sabiranje i množenje. Voditi računa o ispravnosti izraza, odnosno ukoliko se zahteva operacija “skini sa vrha steka” nad praznim stekom, funkcija treba da vrati -1.

14.2 Srednje teški zadaci

1. Napisati program koji proverava da li su etikete u datom HTML fajlu, koji se zadaje, kao prvi argument komandne linije, dobro uparene.(etikete čuvati na steku)
2. Skupovi brojeva su predstavljeni tekstualnim fajlovima koji sadrže brojeve i imena drugih fajlova. Ako se broj x nalazi u fajlu S , tada broj x pripada skupu S . Ako se ime fajla $S1$ nalazi u fajlu $S2$, tada je skup $S1$ podskup skupa $S2$. Program za dati skup S i dati broj x proverava da li x pripada S . (Uputstvo: koristiti red za čuvanje imena fajlova koje tek treba pretražiti da bi se odredilo da li x pripada skupu S)