

1 XI nedelja: Pokazivači na funkcije. Bibliotečke funkcije pretrage i sortiranja.

1.1 Zadaci za rad na času

1. Napisati biblioteku za rad sa jednostruko povezanom listom čiji čvorovi sadrže cele brojeve.
 - (a) Definirati strukturu *Cvor* kojom se predstavlja čvor liste, a koja sadrži ceo broj *vrednost* i pokazivač na sledeći čvor liste.
 - (b) Napisati funkciju *Cvor *napravi_cvor(int broj)* koja kao argument dobija ceo broj, kreira novi čvor liste, inicijalizuje mu polja i vraća njegovu adresu.
 - (c) Napisati funkciju *int dodaj_na_kraj_liste(Cvor ** adresa_glave, int broj)* koja dodaje novi čvor sa vrednošću *broj* na kraj liste. Funkcija treba da vrati 0 ako je sve bilo u redu, odnosno 1 ako je došlo do greške prilikom alokacije memorije za novi čvor.
 - (d) Napisati funkciju *void oslobodi_listu(Cvor ** adresa_glave)* koja oslobađa dinamički zauzetu memoriju za čvorove liste.
 - (e) Napisati funkciju *void ispisi_listu(Cvor * glava)* koja ispisuje vrednosti u čvorovima liste.
 - (f) Napisati funkciju *int velicina_liste(Cvor * glava)* koja vraća broj elemenata liste. Broj elemenata prazne liste je 0.
 - (g) Napisati funkciju *int broj_negativnih(Cvor * glava)* koja vraća broj negativnih elemenata liste.
 - (h) Napisati funkciju *int proizvod_liste(Cvor *glava)* koja računa proizvod svih elemenata liste.
 - (i) Napisati funkciju *int zbir_liste(Cvor * glava)* koja računa zbir svih elemenata liste.
 - (j) Napisati funkciju *float prosek_liste(Cvor * glava)* koja računa prosečnu vrednost svih elemenata liste.
 - (k) Napisati funkciju *Cvor *minimum(Cvor * glava)* koja vraća adresu čvora koji sadrži najmanju vrednost liste. Ukoliko ima više takvih čvorova, vratiti adresu prvog čvora.
 - (l) Napisati funkciju *int dva_najveca(Cvor * glava, int * max1, int * max2)* koja na adrese koje su prosledene kao argumenti funkcije smešta dve najveće vrednosti liste. Ukoliko lista ima manje od 2 čvora funkcija vraća vrednost -1, a inače 0.
 - (m) Napisati funkciju *int uredjena_nerastuce(Cvor * glava)* koja proverava da li su elementi liste uređeni nerastuće. Ukoliko je to slučaj, funkcija treba da vrati vrednost 1, a inače 0.

Napisati program koji koristi jednostruko povezanu listu za čuvanje celih brojeva koji se unose sa standardnog ulaza i testira napisane funkcije. Unošenje novih brojeva u listu prekida se učitavanjem kraja ulaza (EOF).

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
  Unesite elemente liste (Ctrl+D za kraj unosa):
  1 -2 3 8 1 9 7 -8 0
IZLAZ:
  Lista:
  1 -> -2 -> 3 -> 8 -> 1 -> 9 -> 7 -> -8 -> 0
  Broj elemenata liste: 9
  Broj negativnih elemenata liste: 2
  Proizvod elemenata liste: 0
  Zbir elemenata liste: 19
  Prosečna vrednost: 2.11
  Minimum: -8
  Dva najveca elementa: 8 9
  Lista nije uredjena nerastuce.
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
  Unesite elemente liste (Ctrl+D za kraj unosa):
  4 3 2 1 -4
IZLAZ:
  Lista:
  4 -> 3 -> 2 -> 1 -> -4
  Broj elemenata liste: 5
  Broj negativnih elemenata liste: 1
  Proizvod elemenata liste: -96
  Zbir elemenata liste: 6
  Prosečna vrednost: 1.20
  Minimum: -4
  Dva najveca elementa: 3 4
  Lista jeste uredjena nerastuce.
```

2. Definirati strukturu *Cvor* kojom se predstavlja čvor liste koji sadrži karakter *slovo* i pokazivač na sledeći čvor liste, a zatim implementirati sledeće funkcije:
 - (a) *Cvor *napravi_cvor(char slovo)* koja kao argument dobija jedan karakter, kreira novi čvor liste, inicijalizuje mu polja i vraća njegovu adresu.
 - (b) *void oslobodi_listu(Cvor ** adresa_glave)* koja oslobađa dinamički zauzetu memoriju za čvorove liste.
 - (c) *int dodaj_sortirano(Cvor ** adresa_glave, char slovo)* koja dodaje novi element u neopadajuće uređenu listu karaktera, tako da se očuva postojeće uređenje prema vrednosti ASCII koda. Funkcija treba da vrati 0 ako je sve u redu, odnosno 1 ako je došlo do greške prilikom alokacije memorije za novi čvor.
 - (d) *void ispisi_listu(Cvor * glava)* koja ispisuje vrednosti u čvorovima liste bez delimitera (kao jednu nisku).

- (e) *void ispisi_listu_od_pocetka(Cvor * glava)* koja rekurzivno ispisuje vrednosti u čvorovima liste bez delimitera (kao jednu nisku) počevši od prvog elementa liste.
- (f) *void ispisi_listu_od_kraja(Cvor * glava)* koja rekurzivno ispisuje vrednosti u čvorovima liste bez delimitera (kao jednu nisku) počevši od poslednjeg elementa liste.

Napisati program koji koristi jednostruko povezanu listu za čuvanje karaktera koji se unose sa standardnog ulaza i testira rad napisanih funkcija. Unošenje karaktera u listu prekida se učitavanjem karaktera *!*.

Primer 1

```

INTERAKCIJA SA PROGRAMOM:
ULAZ:
  Unesite karaktere (! za kraj unosa):
  j a z d c b k p !
IZLAZ:
  Lista: abcdjkpz
  Lista (rekurzivno, od pocetka): abcdjkpz
  Lista (rekurzivno, od kraja): zpkjdcb

```

Primer 2

```

INTERAKCIJA SA PROGRAMOM:
ULAZ:
  Unesite karaktere (! za kraj unosa):
  b z Z A B c c !
IZLAZ:
  Lista: ABZbccz
  Lista (rekurzivno, od pocetka): ABZbccz
  Lista (rekurzivno, od kraja): zccbZBA

```

1.2 Dodatni zadaci

1. Biblioteku iz prvog zadatka proširiti rekurzivnim implementacijama narednih funkcija:
 - (a) `void ispisi_listu_rekurzivno(Cvor * glava)` koja ispisuje vrednosti u čvorovima liste.
 - (b) `int velicina_liste_rekurzivno(Cvor * glava)` koja vraća broj elemenata liste.
 - (c) `int broj_neparnih_rekurzivno(Cvor * glava)` koja vraća broj neparnih elemenata liste.
 - (d) `int proizvod_liste_rekurzivno(Cvor * glava)` koja računa proizvod svih elemenata liste.
 - (e) `int zbir_liste_rekurzivno(Cvor * glava)` koja računa zbir svih elemenata liste.
 - (f) `int minimum_rekurzivno(Cvor * glava)` koja vraća najmanju vrednost koja se javlja u listi.
 - (g) `int maximum_rekurzivno(Cvor * glava)` koja vraća najveću vrednost koja se javlja u listi.

Primer 1

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
  Unesite elemente liste (Ctrl+D za kraj unosa):
  1 -2 3 8 1 9 7 -8 0
IZLAZ:
Lista:
1 -> -2 -> 3 -> 8 -> 1 -> 9 -> 7 -> -8 -> 0
Broj elemenata liste: 9
Broj neparnih elemenata liste: 5
Proizvod elemenata liste: 0
Zbir elemenata liste: 19
Minimum: -8
Maksimum: 9
```

Primer 2

```
INTERAKCIJA SA PROGRAMOM:
ULAZ:
  Unesite elemente liste (Ctrl+D za kraj unosa):
  -4 3 2 1 -4
IZLAZ:
Lista:
-4 -> 3 -> 2 -> 1 -> -4
Broj elemenata liste: 0
Broj neparnih elemenata liste: 2
Proizvod elemenata liste: 96
Zbir elemenata liste: -2
Minimum: -4
Maksimum: 3
```