

Auditorne vježbe 7.

Teme: strukture, atributi i procedure; pokazivači na strukture; prenošenje struktura kao paramtera (naglasak na prenos po referenci);

7.1 Struktura Vrijeme

Definirajte strukturu “Vrijeme” koja sadrži tri polja “sati”, “minute” i “sekunde”, i koja predstavlja tekuće vrijeme u toku dana, izraženu u satima, minutama i sekundama. Zatim definirajte funkciju “IspisiVrijeme” koja ispisuje vrijeme proslijeđeno kao parametar u funkciju u obliku hh:mm:ss (tj. sati, minute i sekunde se ispisuju kao dvocifreni brojevi, eventualno sa vodećim nulama), kao i funkciju “Saberivrijeme” koja prima dva vremena kao parametre, i vraća kao rezultat treće vrijeme koje nastaje sabiranjem vremena koja su proslijeđena kao parametri (npr. sabiranjem 3h 34min 52sek i 4h 42min 20sek treba da se dobije vrijeme 8h 17min 12sek). Konačno, napravite mali testni program u kojem ćete testirati napisane funkcije.

7.2 Struktura Čvor

Definirajte strukturu “Cvor” koja sadrži polje “element” tipa “double” i polje “veza” koje je tipa pokazivač na “Cvor”. Iskoristite ovu strukturu u programu koji čita slijed realnih brojeva sa tastature sve dok se ne unese 0, uvezuje te elemente u povezanu listu čvorova, i na kraju prolazi kroz listu čvorova sa ciljem da prebroji i ispiše koliko ima elemenata u listi koji su veći od aritmetičke sredine svih unesenih elemenata (potrebna su zapravo dva prolaza: prvi za računanje aritmetičke sredine, i drugi za brojanje elemenata sa traženim svojstvom).

7.3 Struktura Skup

Napisati strukturu “Skup” koja predstavlja skup realnih brojeva. Struktura treba da ima slijedeće parametre: maksimalnu veličinu skupa, stvarnu veličinu skupa i pokazivač na niz tipa double u kojem se čuvaju elementi skupa. Napomenimo da svi elementi u skupu su različiti tj. ako se dodaje neki broj koji je već u skupu, skup ostaje isti.

Potrebno je implementirati slijedeće funkcije koje rade sa strukturom Skup:

```
Skup KreirajSkup(int maksimalnom_velicina);
Skup Unija(Skup s1, Skup s2);
Skup Razlika(Skup s1, Skup s2);
Skup Presjek(Skup s1, Skup s2);
void Ispisi(Skup s) const;
void DodajElement(Skup &s, double element);
```

Funkcija KreirajSkup treba da kreira objekat tipa Skup sa datom maksimalnom veličinom, da postavi stvarnu veličinu na 0, te da alocira dinamički niz sa datom maksimalnom veličinom, te ako dođe problema koda alokacije memorije da se baci izuzetak. Funkcije Unija, Razliku i Presjek trebaju za data dva skupa da izračunaju njihovu uniju, razliku te presjek dva data skupa, te da odgovarajuće vrijednosti vrata kao rezultat. Funkcija Ispisi treba da ispiše dati skup u obliku $\{x_1, x_2, \dots, x_n\}$. Funkcija DodajElement treba datom skupu da doda element pod uslovom da je stvarna veličina skupa manja od maksimalne veličine i da stvarnu veličinu poveća za jedan. Ako nema mjesta potrebno je izvršiti realokaciju novog prostora za elemente skupa kreiranjem novog niza koji je dva puta veći od prethodne veličine, te kopiranjem odgovarajućih elemenata iz starog niza u novi niz i brisanjem starog niza. Nakon ovog se u novi niz dodaje element koji se želi umetnuti.

Potrebno je napraviti testni program u kojem će se demonstrirati upotreba datih funkcija za manipulaciju sa objektima tipa Skup

7.3) Isto kao i 7.2, samo funkcije koje rade sa strukturom *Skup* trebaju izgledati ovako:

```
Skup KreirajSkup(int maksimalnom_velicina);  
Skup& Uniija(const Skup $s1, const Skup $s2);  
Skup* Razlika(const Skup *s1, const Skup *s2);  
Skup* Presjek(const Skup *s1, const Skup *s2);  
void Ispisi(const Skup &s) const;  
void DodajElement(Skup &s, double element);
```

Usporedi brzine izvršavanja ovako definisanih funkcija, sa odgovarajućim funkcijama u 7.2. (Prvo pokušajte sa isključenim opcijama za optimizaciju).