

## Zadaci za 2. laboratorijsku vježbu 2021./2022.

1. Napišite parametriziranu klasu Lista te članske funkcije:

**bool upis (T element) ;**  
**void ispis();**

Funkcija **upis** kao argument prima **element** koji treba dodati u listu tako da lista bude uzlazno sortirana prema vrijednostima elementa. Funkcija vraća true, ako je novi čvor uspješno dodan u listu, a false inače. Funkcija ispis ispisuje cijelu listu od glave.

Pretpostavite da klasa (tip) T ima definirane sve potrebne operatore, konstruktore, destruktore i sl.

Potrebno je definirati strukturu **cvor** koji sadrži element te pokazivač na sljedeći čvor u listi.

U glavnom programu potrebno je s tipkovnice učitati **n** brojeva ( $n \leq 10$ ) te brojeve dodati u listu korištenjem funkcije **upis**. Na kraju je potrebno ispisati članove liste.

2. Napišite parametriziranu klasu Lista te članske funkcije:

**bool upis (string element);**  
**void ispis();**

Funkcija **upis** kao argument prima **element** koji uvijek treba dodati na kraj liste. Funkcija vraća true, ako je novi čvor uspješno dodan u listu, a false inače. Potrebno je definirati strukturu **cvor** koji sadrži string te pokazivače na prethodni i sljedeći čvor u listi.

U glavnom programu potrebno je s tipkovnice učitati **n** stringova ( $n \leq 10$ ) te ih dodati u listu korištenjem funkcije **upis**. Na kraju je potrebno ispisati sve članove liste.

3. Napisati klasu **Stog** koja modelira stog u koji se pohranjuju cijeli brojevi. Stog treba biti implementiran **poljem** i maksimalna veličina mu treba biti **100**. Klasa **Stog** mora (uz potrebne varijable, konstruktore, destruktore i sl.) sadržavati i funkcijski član (metodu) za dodavanje elemenata u stog. U glavnom programu stvoriti jedan stog, nasumično generirati **101** broj te njima napuniti stog.
4. Napisati klasu **Stog** koja modelira stog u koji se pohranjuju cijeli brojevi. Stog treba biti implementiran **pokazivačima**. Klasa **Stog** mora (uz potrebne varijable, konstruktore, destruktore i sl.) sadržavati i funkcijski član (metodu) za dodavanje elemenata u stog. U glavnom programu stvoriti jedan stog, nasumično generirati 101 broj te njima napuniti stog.
5. Napisati klasu **Stog** koja između ostalog (potrebnih varijabli, konstruktora, destruktora i sl.) sadrži i funkcijski član (metodu) za dodavanje i skidanje elemenata sa stoga. Elementi koji se pohranjuju na stog su cijeli brojevi. Stog treba biti implementiran kao polje od 10 članova. U glavnom programu stvoriti stog, nasumično generirati 10 cijelih brojeva iz intervala [1, 10] te njima napuniti stog, a zatim ispisati sadržaj stoga tako da se prvo ispiše element na dnu stoga, a posljednji se ispisuje element na vrhu stoga. Dozvoljeno je korištenje pomoćnog stoga.

6. Strukturu podataka red potrebno je realizirati kao polje. U tu svrhu prvo je potrebno definirati klasu ili strukturu **Red** koja sadrži polje **realnih brojeva** (od najviše 10 članova) te cijele brojeve ulaz i izlaz, koji predstavljaju indekse ulaza i izlaza u red. Napisati potrebne konstruktore, destruktore i sl. Zatim je potrebno napisati funkcijski član (metodu) za dodavanje novog podatka u red te metodu za brisanje podatka iz reda (metode vraćaju **true** ako su uspješno obavile traženu operaciju, a **false** inače):

**bool dodaj (double broj);**  
**bool skini (double \*broj);**

U glavnom programu s tipkovnice učitati **n** realnih brojeva ( $n \leq 10$ ) te ih dodati u red realiziran poljem. Zatim iz reda obrisati sve članove i pri tome ih ispisati. Koristiti metode **dodaj** i **skini**.

7. Strukturu podataka red potrebno je realizirati kao listu, čiji su elementi čvorovi tipa **Cvor**. Klasu/strukturu **Cvor** definirati tako da sadrži realni broj i pokazivač na sljedeći čvor. Također je potrebno definirati klasu **Red** koja sadrži pokazivače ulaz i izlaz, koji pokazuju na ulaz i izlaz iz reda realiziran listom, a čiji su elementi tipa **Cvor**. Napisati potrebne konstruktore, destruktore i sl.

Zatim je potrebno napisati funkcijske članove (metode) za dodavanje novog podatka u red te za dohvat podatka iz reda:

**bool dodaj (double broj);**  
**bool skini (double \*broj);**

U glavnom programu s tipkovnice učitati **n** realnih brojeva ( $n \leq 10$ ) te ih dodati u red realiziran listom pozivom funkcije **dodaj**. Nakon što su svi čvorovi dodani u red, treba iz reda izvaditi jedan po jedan element i ispisati njihove vrijednosti. Koristiti metode **dodaj** i **skini**.

8. Strukturu podataka red potrebno je realizirati kao listu, čiji su elementi čvorovi tipa **Cvor**. Klasu/strukturu **Cvor** definirati tako da sadrži cijeli broj i pokazivač na sljedeći čvor. Također je potrebno definirati klasu **Red** koja sadrži pokazivače ulaz i izlaz, koji pokazuju na ulaz i izlaz iz reda realiziranog listom, a čiji su elementi tipa **Cvor**.

Zatim je potrebno napisati funkcijski član (metodu) za dodavanje novog podatka (**broj**) u red realiziranog listom:

**bool dodaj (double broj);**

Također je potrebno napisati funkcijski član (metodu) koja rekurzivno dodaje elemente polja **polje** od **n** članova u red (pozivati funkciju **dodaj**) tako da se u red prvo doda zadnji element polja. Prototip funkcije je:

**bool poljeURed (int polje[], int n);**

Funkcija treba vratiti **true**, ako je uspjelo dodavanje svih elemenata iz polja u red, a **false**, ako nije uspjelo dodavanje nekog elementa u red. U slučaju neuspješnog dodavanja nekog elementa u red, prekinuti s dodavanjem sljedećih elemenata. Kod dodavanja svakog elementa u red ispisati element koji se upravo dodaje.

U glavnom programu slučajnim odabirom generirati polje **10** cijelih brojeva iz intervala [1, 10] te ih dodati u red realiziran listom pozivom funkcije **poljeURed**. Ispisati članove polja prije dodavanja u red.

9. Napišite funkciju čiji je prototip:

**void insertionSort (Zapis A[], int n, char smjer);**

koja kao argumente prima polje zapisa (**A**) i broj članova polja (**n**) te zastavicu **smjer**. Ako je **smjer** postavljen na 0, onda se ulazni niz sortira uzlazno, a ako je postavljen na 1, niz se sortira silazno. Zapisi u polju sortiraju se algoritmom umetanja (*insertion sort*).

Zapis se sastoji od poštanskog broja mjesta (cijeli broj) i naziva mjesta (string). Sortiranje se obavlja prema poštanskom broju mjesta.

Potrebno je definirati razred ili strukturu **Zapis**, koja sadrži poštanski broj mjesta (cijeli broj) i naziv mjesta (string). U glavnom programu potrebno je s tipkovnice učitati **n** zapisa u polje, gdje je  $n \leq 10$  te smjer sortiranja (0 ili 1). Zatim je potrebno članove polja sortirati pozivom funkcije **insertionSort** te ispisati sortirano polje.

10. Napišite funkciju čiji je prototip:

**void selection2Sort (int A[], int n, char smjer);**

koja kao argumente prima cjelobrojno polje (**A**) i broj članova polja (**n**) te zastavicu **smjer**. Ako je **smjer** postavljen na 0, onda se ulazni niz sortira uzlazno, a ako je postavljen na 1, niz se sortira silazno. Članove polja potrebno je sortirati korištenjem obostranog sortiranja odabirom (*selection sort*).

U glavnom programu potrebno je s tipkovnice učitati **n** članova cjelobrojnog polja, gdje je  $n \leq 10$  te smjer sortiranja (0 ili 1). Zatim je potrebno članove polja sortirati pozivom funkcije **selection2Sort** te ispisati sortirano polje.

11. Napišite funkciju čiji je prototip:

**void obicanBubbleSort (zapis A[], int n, char smjer);**

koja kao argumente prima polje zapisa (**A**) i broj članova polja (**n**) te zastavicu **smjer**. Ako je **smjer** postavljen na 0, onda se ulazni niz sortira uzlazno, a ako je postavljen na 1, niz se sortira silazno. Polje je potrebno sortirati korištenjem običnog *bubble sort*-a.

Zapis se sastoji od šifre jela u restoranu (cijeli broj) i nazivu jela (string). Sortiranje se obavlja prema šifri jela.

Potrebno je definirati razred ili strukturu **zapis** koja sadrži šifru jela (cijeli broj) i naziv jela (string). U glavnom programu potrebno je s tipkovnice učitati **n** zapisa u polje, gdje je  $n \leq 10$  te smjer sortiranja (0 ili 1). Zatim je potrebno članove polja sortirati pozivom funkcije **obicanBubbleSort** te ispisati sortirano polje.

12. Napišite funkciju `InsertionSort` za sortiranje algoritmom Insertion Sort kojim se uzlazno sortiraju **proizvoljni objekti**. Napišite također i klasu `Osoba` koja sadrži podatke o imenu (string) i godinama starosti (unsigned short int) neke osobe, ima potrebne konstruktore, destruktore, gettere i settere, operator pridruživanje te operator „<“ definiran tako da je „manja“ ona osoba koja ima manje godina, a ako imaju isti broj godina, onda je manja ona čije ime počinje na „manje“ slovo u abecedi (npr. ako postoje objekti {„Ivo“,9}, {„Marko“,9}, manji je {„Ivo“,9}). Napišite glavni program u kojem se stvara polje od 10 objekata tipa `Osoba`, poziva funkcija za sortiranje te ispisuju rezultati sortiranja. Funkcija mora imati prototip:

**void InsertionSort(T A[], int N);**

Primjer: Za ulazno polje {{„Ana“,20},{„Ivo“,9},{„Marko“,9},{„Lidija“,22},{„Pero“,19}}, algoritam mora kao rezultat vratiti sortirano polje: {{„Ivo“,9},{„Marko“,9},{„Pero“,19},{„Ana“,20},{„Lidija“,22}}.

13. Napišite funkciju `SelectionSort` za sortiranje algoritmom Selection Sort kojim se uzlazno sortiraju **proizvoljni objekti**. Napišite također i klasu `Vozilo` koja sadrži podatke o modelu vozila (string) i godini proizvodnje(int), ima potrebne konstruktore, destruktore, gettere i settere, operator pridruživanje te operator „<“ definiran tako da je „manje“ vozilo koje je abecedno „manje“, a ako su modeli isti, onda je manje ono koje je „mlađe“, npr. za vozila {„Pauegot“,1983} i {„Pauegot“,1981} manje je mlađe vozilo, tj. {„Peugeot“,1983}. Napišite glavni program u kojem se stvara polje od 10 objekata tipa `Vozilo`, poziva funkcija za sortiranje te ispisuju rezultati sortiranja. Funkcija mora imati prototip:

**void SelectionSort(T A[], int N);**

U sklopu realizacije potrebno je izraditi i koristiti funkciju prototipa:

**void Zamijeni(vozilo &prvi, vozilo &drugi);**

Primjer: Za ulazno polje {{„Pauegot“,1981},{„Pauegot“,1983},{„Ranulet“,1967},{„Fait“,1972},{„BWM“,1985},{„Merdesec“,1983}}, algoritam mora kao rezultat vratiti sortirano polje: {{„BWM“,1985},{„Fait“,1972},{„Merdesec“,1983},{„Pauegot“,1983},{„Pauegot“,1981},{„Ranulet“,1967}}.

14. Napišite funkciju `BubbleSortPoboljsani` za sortiranje algoritmom Bubble Sort (poboljšana verzija) kojim se sortiraju **stringovi** i glavni program u kojem se rezervira memorija za 10 proizvoljnih stringova, poziva funkcija za sortiranje te ispisuju rezultati sortiranja.

Funkcija mora imati prototip:

**void BubbleSortPoboljsani(string A[], int N, int smjer);**

U sklopu realizacije potrebno je izraditi i koristiti funkciju prototipa:

**void Zamijeni(string \*prvi, string \*drugi);**

Primjer: Za ulazno polje {„Ivo“, „Marko“, „Juraj“, „Pero“} i za uzlazni smjer sortiranja (smjer = 1), algoritam mora kao rezultat vratiti sortirano polje znakovnih nizova: {„Ivo“, „Juraj“, „Marko“, „Pero“}. Napomena: silazni smjer sortiranja se funkciji šalje kao smjer = 0.