

Zadatci za 2. laboratorijske vježbe 2016./2017.

1. Napišite funkciju čiji je prototip:

```
void insertionSort (zapis *A, int n, char smjer);
```

koja kao argumente prima pokazivač na početak polja 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 (40+1 znakova). Sortiranje se obavlja prema poštanskom broju mjesta.

Potrebno je definirati strukturu **zapis** koji sadrži poštanskog broja mjesta (cijeli broj) i naziva mjesta (40+1 znakova). 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.

2. Napišite funkciju čiji je prototip:

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

koja kao argumente prima pokazivač na početak cjelobrojnog polja (**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.

3. Napišite funkciju čiji je prototip:

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

koja kao argumente prima pokazivač na početak polja 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 (50+1 znakova). Sortiranje se obavlja prema šifri jela.

Potrebno je definirati strukturu **zapis** koji sadrži šifru jela (cijeli broj) i naziv jela (50+1 znakova). 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.

4. Napišite funkciju čiji je prototip:

```
void spojiPolja(int *A, int *B, int *Rez, int n1, int n2);
```

koja kao argumente prima tri pokazivača (**A**, **B** i **Rez**), gdje svaki pokazivač pokazuje na početak cjelobrojnog polja te broj članova polja u poljima na koje pokazuju **A** i **B** (**n1** i **n2**). Polja na koje

pokazuju pokazivači **A** i **B** su silazno sortirana. Polje na koje pokazuje **Rez** treba napuniti članovima polja **A** i **B** tako da članovi toga polja budu silazno sortirani po izlasku iz funkcije. Pretpostaviti da u polju na koje pokazuje **Rez** ima dovoljno mjesta za pohraniti sve elemente iz polja na koje pokazuju **A** i **B**.

U glavnom programu definirati **A** s elementima [9, 7, 5, 3, 1] i **B** s elementima [8, 6, 4, 2, 0]. Rezervirati memoriju za novo polje **Rez**, koje treba napuniti elementima iz **A** i **B** korištenjem funkcije **spojiPolja**. Na kraju ispisati elemente polja **Rez**.

5. Napišite funkciju čiji je prototip:

```
void spojiNizove(char *A, char *B, char *Rez);
```

koja kao argumente prima tri pokazivača (**A**, **B** i **Rez**), gdje svaki pokazivač pokazuje na početak znakovnog niza (*string*-a). Znakovi u znakovnim nizovima na koje pokazuju pokazivači **A** i **B** su silazno sortirani prema ASCII vrijednosti znaka. Znakovni niz na koje pokazuje **Rez** treba napuniti znakovima iz znakovnih polja **A** i **B** tako da znakovi budu silazno sortirani prema ASCII vrijednosti po izlasku iz funkcije. Pretpostaviti da znakovnom nizu na koje pokazuje **Rez** ima dovoljno mjesta za pohraniti sve znakove iz znakovnih polja na koje pokazuju **A** i **B**.

U glavnom programu definirati znakovno polje **A** ("ecaDB") i znakovno polje **B** ("dbECA"). Rezervirati memoriju za novi znakovni niz **Rez**, koje treba napuniti znakovima iz **A** i **B** korištenjem funkcije **spojiNizove**. Na kraju ispisati znakovni niz **Rez**.

6. Napišite funkciju čiji je prototip:

```
int upis (cvor **glava, int element) ;
```

koja kao argumente prima adresu pokazivača na početak liste (**glava**) i cijeli broj (**element**) koji treba dodati u listu tako da lista bude uzlazno sortirana prema vrijednostima cijelih brojeva. Funkcija vraća 1, ako je novi čvor uspješno dodan u listu, a 0 inače.

Potrebno je definirati strukturu **cvor** koji sadrži cijeli broj 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.

7. Napišite funkciju čiji je prototip:

```
int upis(cvor **glava, char *niz);
```

koja kao argumente prima adresu pokazivača na početak liste (**glava**) i znakovni niz (**niz**), koji uvijek treba dodati na početak liste. Funkcija vraća 1, ako je novi čvor uspješno dodan u listu, a 0 inače.

Potrebno je definirati strukturu **cvor** koji sadrži znakovni niz bez praznina (40 + 1 znak) te pokazivač na sljedeći čvor u listi.

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

8. Napišite funkciju InsertionSort za sortiranje algoritmom Insertion Sort kojim se sortiraju **znakovni nizovi** i glavni program u kojem se rezervira memorija za 10 proizvoljnih znakovnih nizova, poziva funkcija za sortiranje te ispisuju rezultati sortiranja.

Funkcija mora imati prototip:

```
void InsertionSort(char **A, int N, int smjer);
```

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.

9. Napišite funkciju SelectionSort za sortiranje algoritmom Selection Sort kojim se sortiraju **znakovni nizovi** i glavni program u kojem se rezervira memorija za 10 proizvoljnih znakovnih nizova, poziva funkcija za sortiranje te ispisuju rezultati sortiranja.

Funkcija mora imati prototip:

```
void SelectionSort(char **A, int N, int smjer);
```

U sklopu realizacije potrebno je izraditi i koristiti funkciju prototipa:

```
void Zamijeni(char *prvi, char *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.

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

Funkcija mora imati prototip:

```
void BubbleSortPoboljsani(char **A, int N, int smjer);
```

U sklopu realizacije potrebno je izraditi i koristiti funkciju prototipa:

```
void Zamijeni(char *prvi, char *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.

11. Napišite funkciju `ShellSort` za sortiranje algoritmom Shell Sort kojim se sortiraju znakovni nizovi. U Shell sortu je potrebno koristiti niz koraka koji odgovara skupu $\{N/2, N/4, \dots, 1\}$, pri čemu je N broj elemenata polja (tj. broj znakovnih nizova u polju). Potrebno je napisati i glavni program u kojem se rezervira memorija za 10 proizvoljnih znakovnih nizova, poziva funkcija za sortiranje te ispisuju rezultati sortiranja.

Funkcija mora imati prototip:

```
void ShellSort(char **A, int N, int smjer);
```

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 $smjer = 0$.

12. Napišite funkciju `ShellSort_Hibbard` za sortiranje algoritmom Shell Sort kojim se sortiraju znakovni nizovi. U Shell sortu je potrebno koristiti niz koraka koji odgovara Hibbardovom slijedu ($\{1, 3, 7, \dots, 2^k - 1\}$ – detalje pogledati u predavanjima). Potrebno je napisati i glavni program u kojem se rezervira memorija za 10 proizvoljnih znakovnih nizova, poziva funkcija za sortiranje te ispisuju rezultati sortiranja.

Funkcija mora imati prototip:

```
void ShellSort_Hibbard(char **A, int N, int smjer);
```

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 $smjer = 0$.

Napomena: Odabir koraka mora biti takav da počne s Hibbardovim brojem koji je prvi manji od dimenzije polja, kojeg se zatim smanjuje do koraka s vrijednosti 1. Npr. Za $N=12$, bira se slijed koraka $\{7, 3, 1\}$, budući da je 7 prvi Hibbardov broj manji od $N=12$.