

PROGRAMIRANJE II (08.09.2020.)

- ❶ (15 bodova) Neka je dat tip:

```
typedef struct {  
    char prezime[21], ime[21];  
    int godine_starosti;  
} OSOBA;
```

Napisati funkciju sa promjenljivim brojem argumenata, koja koristeći *selection_sort* algoritam vrši sortiranje osoba po nekom kriterijumu.

Prototip funkcije je sljedeći:

```
OSOBA *custom_sort(int (*cmp)(OSOBA *,  
                                OSOBA *), int n, ...);
```

Funkcija kao argumente prima funkciju za poređenje dvije osobe po nekom kriterijumu, broj osoba, te osobe koje je potrebno sortirati (kao neobavezne argumente). Funkcija vraća sortirani dinamički niz osoba.

U glavnoj funkciji (*main*) potrebno je demonstrirati sortiranje proizvoljnog broja osoba u opadajućem redoslijedu po godinama starosti koristeći funkciju *custom_sort*.

- ❷ (15 bodova) Neka je dat tip za reprezentaciju dolaznog telefonskog poziva:

```
typedef struct dolazni_poziv {  
    char *broj_pozivaoca; // broj pozivaoca  
    int trajanje; // trajanje poziva u sekundama  
} D_POZIV;
```

Neka je dat tip za reprezentaciju telefonskog pretplatnika:

```
typedef struct pretplatnik {  
    char *broj; // broj telefona  
    D_POZIV *poziv; // niz dolaznih poziva  
    int n; // broj dolaznih poziva  
} PRETPLATNIK;
```

Napisati program koji iz tekstualne datoteke učitava podatke o pozivima te za svakog pretplatnika ispisuje na standardnom izlazu dolazne pozive sa broja koji je proslijeđen kao argument komandne linije. Niz tipa *PRETPLATNIK* mora biti dinamički. Pretpostaviti da su svi telefonski brojevi u istom formatu (isti broj cifara). Primjer tekstualne datoteke dat je u nastavku (prva kolona je broj koji je pozivan, druga kolona je broj pozivaoca, a treća kolona trajanje poziva u sekundama):



brojevi.txt - Notepad

```
File Edit Format View Help  
065/111-111 065/222-222 80  
065/333-333 065/444-444 60  
065/111-111 065/222-222 70  
065/444-444 065/111-111 90  
065/333-333 065/111-111 120
```

- ❸ (15 bodova) Definirati tip *CVOR* kojim se reprezentuje čvor jednostruko povezane liste, pri čemu je informacioni sadržaj cjelobrojnog tipa.

Napisati funkciju kojom se elementi niza, koji se proslijeđuje kao argument funkcije, dodaju na kraj liste. Prototip funkcije je:

```
void addAll(CVOR **plista, int br_el_niza,  
            int *niz);
```

Napisati funkciju kojom se elementi niza, koji se proslijeđuje kao argument funkcije, brišu iz liste. Prototip funkcije je:

```
void removeAll(CVOR **plista, int br_el_niza,  
               int *niz);
```

Napisati glavnu funkciju (*main*) u kojoj se demonstrira upotreba funkcija *addAll* i *removeAll* na dinamički kreiranom nizu elemenata.

Napomena: broj elemenata dinamičkog niza i elementi niza učitavaju se putem standardnog ulaza.

- ❹ (15 bodova) Neka je dat tip:

```
typedef struct node_t {  
    int info; // informacioni sadrzaj  
    struct node_t *left, *right;  
} NODE_T;
```

kojim se reprezentuje čvor stabla binarne pretrage, pri čemu je informacioni sadržaj čvora cjelobrojni podatak (*info*).

Definirati tip *GRAF* koristeći matričnu reprezentaciju grafa, pri čemu je informacioni sadržaj svakog čvora grafa pokazivač na korijen binarnog stabla pretrage (tip *NODE_T*).

Napisati funkciju koja računa i kao rezultat vraća sumu grafa. Suma grafa predstavlja sumu informacionih sadržaja (podatak *info*) svih čvorova binarnih stabala pretrage koja se nalaze u grafu. Prototip funkcije je:

```
int suma(GRAF *graf);
```

Napomena: Pretpostaviti da je graf povezan. Posjetu svakom od čvorova grafa vršiti obilaskom grafa po dubini (DFS).