

## PROGRAMIRANJE II (27.10.2020.)

- ❶ (15 bodova) Neka je definisan tip:

```
typedef struct sjediste
{
    char *sifra;
} SJEDISTE;
```

kojim se reprezentuje sjedište autobusa i neka je definisan tip:

```
typedef struct autobus
{
    char *naziv;
    SJEDISTE *sjedista;
    int broj_sjedista;
} AUTOBUS;
```

kojim se reprezentuje autobus koji u sebi ima adresu početka dinamičkog niza sjedišta *sjedista* čija je dužina definisana promjenljivom *broj\_sjedista*.

Napisati funkciju sa promjenljivim brojem argumenata čiji je protip:

```
SJEDISTE* sjedista(int *broj, int n,...);
```

Neobavezni argumenti su tipa AUTOBUS. Funkcija iz svakog autobusa preuzima sjedišta, kreira dinamički niz sjedišta te ih sortira selection sort algoritmom alfanumerički po šifri (*sifra*) u rastućem redoslijedu. Adresa početka niza je promjenljiva *sjedista* a broj elemenata niza promjenljiva *broj*.

- ❷ (15 bodova) U ulaznoj binarnoj datoteci, u prva četiri bajta upisana je informacija o broju podataka koji su upisani u datu datoteku, dok je u sljedećem bajtu upisana informacija o veličini jednog upisanog podatka. Nakon toga su upisani podaci.

Napisati funkciju koja podatke upisane u datoteku učitava u memoriju (*heap*) i vraća adresu početka podataka u memoriji, dok informacije o broju podataka i veličini jednog podatka vraća preko parametara *nmem* (broj podataka) i *size* (veličina jednog podatka). Prototip funkcije je:

```
void* d_read(FILE *f,
    unsigned *nmem, unsigned char *size);
```

U glavnom programu (korištenjem funkcije *d\_read*) pročitati podatke iz datoteke čiji je naziv prvi argument komandne linije. Pod pretpostavkom da su u ulaznoj datoteci upisani označeni cjelobrojni podaci date veličine, sekvencijalno pretražiti pročitane podatke i ispisati informaciju o tome da li se među učitanim podacima nalazi broj -1.

- ❸ (15 bodova) Neka su dati sljedeći tipovi:

```
typedef struct node1 {
    char *rijec;
    struct node1 *next;
} NODE1;
```

```
typedef struct node2 {
    char *rijec;
    struct node2 *right, *left;
} NODE2;
```

koje predstavljaju čvorove stabla jednostruko i dvostruko povezane liste respektivno. Napisati funkciju koja informacioni sadržaj jednostruko povezane liste kopira u dvostruko povezanu listu. Pri tome je potrebno kopirati sadržaj svakog drugog čvora počevši od prvog čvora jednostruko povezane liste. Sve dodatne pomoćne funkcije, u slučaju korištenja, potrebno je definisati.

- ❹ (15 bodova) Neka je zadato binarno stablo koje se sastoji od maksimalno *n* čvorova sa informacionim sadržajem tipa VOZILO:

```
typedef struct vozilo
{
    char broj_sasije[8], boja[21];
    double snaga;
} VOZILO;
```

Definisati tip *NODE* (čvor binarnog stabla pretrage) u kojem je informacioni sadržaj tip *VOZILO*. Ključ za dodavanje u binarno stablo pretrage je *broj\_sasije*. Definisati tip *GRAF* (matrična reprezentacija) u kojem je informacioni sadržaj tip *VOZILO*. Napisati funkciju koja binarno stablo konvertuje u strukturu neusmjerenog grafa koji maksimalno može da sadrži *n* čvorova. Nakon toga napisati funkciju koja prolazi kroz sve čvorove grafa i upisuje ih na standardni izlaz. U glavnom programu demonstrirati konverziju binarnog stabla u graf i ispis podataka na standardni izlaz. Za binarno stablo se smatra da je već kreirano te se u funkciju za konverziju proslijeđuje korijen stabla. Stablo obilaziti u pre-order redoslijedu obilaska.