

PROGRAMIRANJE II (25.08.2020)

- ❶ (15 bodova) Napisati funkciju sa promenljivim brojem argumenata koja formira i vraća dinamičku matricu realnih brojeva tipa *double*. Funkcija prihvata obavezne parametre *n* i *m* koji predstavljaju dimenzije matrice, a zatim *nxm* realnih brojeva tipa *double* koji predstavljaju elemente matrice (elementi su navedeni redom po redovima). Prototip funkcije je:

```
double** mat_init(int n, int m, ...);
```

Napisati funkciju koja iz ulazne tekstualne datoteke čiji je naziv parametar *f* učitava matricu u (dinamičku) memoriju. Ulazna datoteka formatirana je tako da je u prvom redu upisan broj redova, u drugom redu upisan je broj kolona, a nakon toga upisani su elementi matrice (u svakoj liniji upisani su elementi jednog reda međusobno razdvojeni blanko znakom). Pretpostaviti da je ulazna datoteka (ako postoji) pravilno formatirana. U slučaju da ulazna datoteka ne postoji, funkcija treba da vrati NULL. Preko parametara *n* i *m* funkcija treba da vrati dimenzije učitane matrice. Prototip funkcije je:

```
double** mat_read(const char *f, int *n, int *m);
```

- ❷ (15 bodova) Neka je dat tip:

```
typedef struct foto {  
    char naziv[64];  
    int v; // velicina u bajtovima  
} FOTO;
```

kojim se reprezentuju informacije o jednoj fotografiji.

Napisati funkciju koja niz od *n* informacija o fotografijama sortira po nazivu u rastućem redosledu korištenjem *shell-sort* algoritma, a čiji je prototip:

```
void sh_sortiraj(FOTO *niz, int n);
```

Napisati nerekurzivnu funkciju koja vrši binarno pretraživanje (po nazivu) niza informacija o fotografijama. Funkcija vraća adresu pronađenog podatka ili NULL ako fotografija sa datim nazivom ne postoji. Prototip funkcije je:

```
FOTO* b_trazi(FOTO *niz, int n,  
              const char *naziv);
```

- ❸ (15 bodova) Neka je dat tip:

```
typedef struct cvor {  
    double d; // inf. sadrzaj  
    struct cvor *sledeci;  
} CVOR;
```

kojim se reprezentuje čvor jednostruko ulančane liste realnih brojeva.

Napisati funkciju koja dodaje novi broj na kraj liste, a čiji je prototip:

```
void dodaj(CVOR **pglava, double d);
```

Napisati funkciju (korištenjem funkcije *dodaj*) koja formira i vraća dinamički niz od *n* jednostruko povezanih listi (svaki element niza je pokazivač na prvi čvor liste) na osnovu matrice dimenzija *nxm*, pri čemu elementi *k*-te liste treba da odgovaraju elementima *k*-tog reda matrice. Prototip funkcije je:

```
CVOR** mat2arr(double **mat, int n, int m);
```

- ❹ (15 bodova) Neka je dat tip:

```
typedef struct graf {  
    int n; // broj cvorova  
    char inf[MAX]; // inf. sadrzaj cvorova  
    int ms[MAX][MAX]; // matrica susednosti  
} GRAF;
```

kojim se reprezentuje neusmeren graf pri čemu informacioni sadržaj čvora grafa predstavlja jedan znak.

Napisati funkciju koja kreira i vraća (dinamički) graf koji predstavlja stablo razapinjanja grafa koji je parametar funkcije. Stablo razapinjanja treba formirati na osnovu obilaska grafa po dubini počevši od čvora sa indeksom 0, pri čemu obilazak treba implementirati u zasebnoj rekurzivnoj funkciji (koja može da bude i ugneždena). Takođe, prilikom formiranja stabla razapinjanja potrebno je redom ispisivati grane koje se dodaju. Svaka grana se ispisuje u zasebnom redu u formatu *cvor1->cvor2*, gde je *cvor1* informacioni sadržaj polaznog, a *cvor2* informacioni sadržaj dolaznog čvora. Prototip funkcije je:

```
GRAF* st(const GRAF *g);
```