

PROGRAMIRANJE II – K2
(08.06.2022)

❶ (10 bodova) Neka je dat tip:

```
typedef struct nastavnik {  
    char *ime, *prezime;  
} NASTAVNIK;
```

kojim se reprezentuje tip nastavnik.

Neka je dat tip:

```
typedef struct predmet {  
    char *naziv;  
    NASTAVNIK *niz;  
    int n;  
} PREDMET;
```

kojim se reprezentuje tip predmet, pri čemu je *niz* adresa početka niza nastavnika, a *n* broj elemenata niza.

Neka je dat tip:

```
typedef struct node {  
    PREDMET predmet;  
    struct node *left, *right;  
} NODE;
```

kojim se reprezentuje čvor dvostruko povezane liste čiji je informacioni sadržaj predmet.

Napisati funkciju koja briše iz dvostruko povezane liste (parametri *head* i *tail*) sve čvorove koji ispunjavaju određeni uslov. Čvorovi za brisanje ispunjavaju uslov ukoliko pokazivač na funkciju (parametar *criteria*) koji je proslijeđen kao argument funkcije vraća vrijednost 1 za trenutni čvor. Prototip funkcije je:

```
void delete(NODE **head, NODE **tail, int  
            (*criteria)(NODE *n))
```

❷ (10 bodova) Izabrati i definisati najpogodnije tipove za rješavanje sljedećeg problema:

Studenti čekaju u koloni da dobiju ulaznicu za studentsku predstavu. Postoje dvije vrste predstave i na ulaznicama su označene oznakama 1 i 2. Ulaznice se nalaze u kutiji. Svaki student želi da gleda određenu predstavu i uzima ulaznicu sa vrha kutije. Ukoliko se oznaka ulaznice poklapa sa predstavom koju student želi da gleda, student uzima ulaznicu i napušta kolonu. U suprotnom, ne uzima ulaznicu (ostavlja je na vrhu kutije), a zatim se vraća na kraj kolone.

Napisati funkciju koja računa koliko studenata će otići da gleda predstavu. Funkcija treba da vrši računanje sve dok svi studenti ne uzmu ulaznicu ili dok broj vraćanja ulaznice u kutiju ne bude jednak broju *n* proslijeđenom kao parametar funkcije.

Koristiti ulančane reprezentacije odabranih tipova. Smatrati da je jednak broj studenata i ulaznica.

❸ (10 bodova) Neka je dat tip:

```
typedef struct node {  
    int info;  
    struct node *left, *right;  
} NODE;
```

kojim se reprezentuje čvor binarnog stabla, čiji je informacioni sadržaj cijeli broj.

Napisati rekurzivnu funkciju koja provjerava da li je zadato binarno stablo (parametar *root*) stablo binarnog pretraživanja. Funkcija treba da vrati 1 ako jeste stablo binarnog pretraživanja, a 0 ako nije stablo binarnog pretraživanja.

Prototip funkcije je:

```
int is_bst(NODE *root);
```

❹ (10 bodova) Neka je dat tip:

```
typedef struct node {  
    int info;  
    struct node *next;  
} NODE;
```

kojim se reprezentuje čvor jednostruko ulančane liste.

Neka je dat tip:

```
typedef struct graph {  
    int n; // broj cvorova  
    NODE **al; // liste susjednosti tj. niz od  
               // n ulančanih listi  
} GRAPH;
```

koji predstavlja ulančanu reprezentaciju usmjerenog grafa kod kojeg liste susjednosti ne moraju biti uređene.

Napisati funkciju koja određuje i vraća broj susjeda čvora čiji je indeks *u*. Prototip funkcije je:

```
int broj_susjeda(GRAPH *g, int u);
```

Napisati funkciju koja dodaje novu granu između čvorova čiji su indeksi *u* i *v*. Ako ta grana već postoji u grafu, ignorisati pokušaj dodavanja grane. Prototip funkcije je:

```
void dodaj_granu(GRAPH *g, int u, int v);
```