

PROGRAMIRANJE II – K2  
(11.06.2021)

❶ (5 bodova) Neka je dat tip:

```
typedef struct node {  
    char *val;  
    struct node *next;  
} NODE;
```

kojim se reprezentuje čvor jednostruko povezane liste čiji je informacioni sadržaj string.

Neka je dat tip:

```
typedef struct double_node {  
    char *val;  
    struct double_node *left, *right;  
} DOUBLE_NODE;
```

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

Napisati funkciju koja briše iz jednostruko povezane liste (parametar *node*) sve čvorove koji ispunjavaju određeni uslov. Funkcija kao rezultat vraća adresu glave novokreirane dvostruko povezane liste u čije čvorove je upisan informacioni sadržaj obrisanih čvorova. Čvorovi za brisanje ispunjavaju uslov ukoliko pokazivač na funkciju (parametar *criteria*) koji je proslijeđen kao argument funkcije vraća vrijednost 1. Prototip funkcije je:

```
DOUBLE_NODE* delete(NODE **node, int  
    (*criteria)(NODE *n))
```

❷ (5 bodova) Odabrati najadekvatniju strukturu za implementaciju *undo* akcije. *Undo* akcija podrazumijeva poništenje efekta prethodne akcije.

Implementirati kreiranje akcije u vidu upisivanja cijelog broja u datu strukturu, dok poništavanje akcije odnosno *undo* podrazumijeva uklanjanje broja iz strukture. Maksimalan broj akcija nije unaprijed poznat.

Predstaviti sadržaj strukture nakon izvršenja sljedećih akcija:

Dodavanje: 1

Dodavanje: 2

Undo

Dodavanje: 3

Undo

Dodavanje: 5

Dodavanje: 6

Undo.

❸ (5 bodova) Neka je dat tip:

```
typedef struct node {  
    char str[64];  
    struct node *left, *right;  
} NODE;
```

kojim se reprezentuje čvor stabla binarne pretrage stringova. Napisati rekurzivnu funkciju koja provjerava i vraća informaciju o tome da li se u stablu binarne pretrage čiji je korijen prvi parametar funkcije, nalazi string koji je drugi parametar funkcije. Funkcija treba da vrati vrijednost 1 ako se traženi string nalazi u stablu, a u suprotnom 0.

Pri tome, na standardni izlaz potrebno je ispisati informacioni sadržaj svih posjećenih čvorova tj. sve stringove koji su poređeni sa ključem pretrage (svaki string se ispisuje u zasebnom redu). Takođe, ako traženi string postoji u stablu, tada je nad pronađenim stringom potrebno primijeniti funkciju čija je adresa treći parametar funkcije *find*.

Prototip funkcije je:

```
int find(NODE *root, const char *key,  
    void (*)(const char *s));
```

❹ (5 bodova) Neka je dat tip:

```
typedef struct {  
    int n; // broj cvorova  
    char znakovi[100]; // inf. sadrzaj  
    int ms[100][100];  
} GRAF;
```

kojim se reprezentuje usmjeren graf sa najviše sto čvorova. Napisati funkciju koja za zadati graf *g* (parametar funkcije) vraća redni broj čvora sa indeksom *kraj* (parametar funkcije) prilikom obilaska grafa po dubini (*DFS*), pri čemu obilazak treba da započne od čvora čiji je indeks *start* (parametar funkcije). Ukoliko čvor sa indeksom *kraj* nije dostižan pri *DFS* obilasku počevši od čvora sa indeksom *start*, funkcija treba da vrati 0. Prototip funkcije je:

```
int dfs_rb(GRAF *g, int start, int kraj);
```