ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMIRANJE II – K2
(07.06.2023)

❶ **(10 bodova**) Neka su dati tipovi:

```
typedef struct glumac {
    char *ime, *prezime;
} GLUMAC;
typedef struct film {
    GLUMAC *niz;
    int n; //broj glumaca u filmu
} FILM;
```

kojim se reprezentuju glumac i film u kojem glume glumci, respektivno.

Neka su dati tipovi:

```
typedef struct glumac_pojavljivanja {
    char *ime, *prezime;
    int broj_pojavljivanja;
} GLUMAC_POJAVLJIVANJA;
typedef struct cvor {
    GLUMAC_POJAVLJIVANJA gp;
    struct cvor *sljedeci;
} CVOR;
```

kojim se reprezentuju glumac sa ukupnim brojem pojavljivanja u svim filmovima i čvor jednostruko povezane liste čiji je informacioni sadržaj glumac sa brojem pojavljivanja u filmovima, respektivno.

Napisati funkciju koja iz niza filmova kreira jednostruko povezanu listu u čijim su čvorovima informacioni sadržaj glumci (tip *GLUMAC_POJAVLJIVANJA*) sa ukupnim brojem pojavljivanja u svim filmovima. Funkcija kao rezultat vraća glavu liste.

Glumac se jedinstveno identifikuje imenom i prezimenom (u obzir uzeti velika i mala slova). Prototip funkcije je:

CVOR* dodaj(FILM *f, int n);

Pri tome je parametar *f* adresa niza filmova, a parametar *n* broj filmova u nizu.

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

Korisnik sluša pjesme iz određene plejliste. Tokom slušanja plejliste korisnik koristi akcije za prebacivanje pjesama naprijed i nazad. Napisati funkciju koja prima adekvatnu strukturu pjesama i listu akcija te na osnovu datih podataka pronaći koja je posljednja pjesma koju je korisnik slušao. Akcije treba da budu definisane tipom int gdje broj 0 definiše prebacivanje unazad, a broj 1 prebacivanje unaprijed. Ukoliko se dođe do kraja plejliste, potrebno je ponovo krenuti od prve pjesme.

Koristiti ulančane reprezentacije odabranih tipova.

❸ **(10 bodova)** Neka su dati tipovi

```
typedef struct point {
    double x, y;
} POINT;
typedef struct tn {
    POINT p;
    struct tn *l, *r;
} TREE_NODE;
```

kojim se reprezentuju tačka u dvodimenzionom prostoru i čvor binarnog stabla tačaka, respektivno.

Napisati rekurzivnu funkciju koja dodaje novu tačku (parametar *p*) u stablo binarne pretrage tačaka prema nekom kriterijumu. Funkcija prihvata (parametar *root*) i vraća korijen stabla. Kriterijum pretrage je funkcija na koju pokazuje *cmp*, a koja vraća rezultat poređenja dvije tačke, i to: -1 ako je prva tačka manja od druge, 0 ako su tačke jednake i 1 ako je prva tačka veća od druge. Prototip funkcije je:

TREE_NODE* bst_add(TREE_NODE *root, const POINT *p, int (*cmp)(const POINT *p1, const POINT *p2));

Napisati rekurzivnu funkciju koja u izlaznu tekstualnu datoteku (parametar *f*) upisuje stablo binarne pretrage tačaka u rastućem redoslijedu. Korijen stabla je parametar *root*. Svaku tačku treba upisati u zaseban red u formatu (x,y). Prototip funkcije je:

void bst_print(TREE_NODE *root, FILE *f);

❹ **(10 bodova)** Neka je dat tip:

```
typedef struct grad {
    char naziv[101];
} GRAD;
```

kojim se reprezentuju podaci o gradu.

Neka je dat tip:

```
typedef struct graf {
    int n;                  // broj čvorova
    GRAD gradovi[100];      // informacioni
                            //sadržaj
    int ms[100][100];       //matrica
                            //susjednosti
                            //težinskog,
                            //neusmjerenog
                            //grafa
} GRAF;
```

kojim se reprezentuje neusmjeren težinski graf sa najviše stotinu čvorova kojim su predstavljeni gradovi i njihova povezanost (rastojanje u km).

Napisati funkciju koja za dati graf *gf* i grad *gr* ispisuje najkraće rastojanje od grada *gr* do svih dostupnih gradova. Za svaki dostupni grad ispisati naziv i koja je najkraća udaljenost od zadatog grada.

Prototip funkcije je:

void dostupni(GRAF gf, GRAD gr);