ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMIRANJE II (11.09.2018.)

❶ (**25 bodova**) Neka je definisan tip:

```
typedef struct vagon
{
   char *naziv;
} VAGON;
```

kojim se reprezentuje vagon voza i neka je definisan tip:

```
typedef struct voz
{
   char *naziv;
   VAGON *vagoni;
   int brojVagona;
} VOZ;
```

kojim se reprezentuje voz koji u sebi ima adresu početka dinamičkog niza vagona *vagoni* čija je dužina definisana promjenljivom *brojVagona*.

Napisati funkciju sa promjenljivim brojem argumenata čiji je protitip:

```
void vagoni(VAGON **vagoni, int *broj, int n,...);
```

Neobavezni argumenti su tipa VOZ. Funkcija iz svakog voza preuzima vagone, kreira dinamički niz vagona te ih sortira insertion sort algoritmom po nazivu vagona alfanumerički u opadajućem redoslijedu. Pretpostaviti da niz nije bio dinamički alociran prije dodavanja vagona u isti. Adresa početka niza je promjenljiva *vagoni* a broj elemenata niza promjenljiva *broj*.

❷ (**25 bodova**) Napisati program koji simulira pretragu baze osoba. Potrebno je u datoteci `baza.txt` pronaći podatke o osobi (ime, prezime, jmbg i adresa stanovanja), čiji se jmbg unosi sa standardnog ulaza (nije potrebna validacija jmbg-a). Pri tome važi da je jmbg jedinstven. Potrebno je iz datoteke `jmbg_indeksi.txt` pročitati sve indeksne zapise (look-up tabela) i formirati dinamički niz, u kojem će se na osnovu unesenog jmbg-a (ključ), pronaći odgovarajuća adresa početka zapisa o osobi u datoteci `baza.txt`, te na taj način pročitati i ispisati podaci o pronađenoj osobi. Za pretragu datoteke `jmbg_indeksi.txt` koristiti sekvencijalnu pretragu sa stražom. Prilikom poređenja dva jmbg-a napisati i koristiti pomoćnu funkciju koja će kao argument primiti pokazivač na funkciju za poređenje. Uz pomoć prethodno realizovanih funkcija definisati glavni program i ilustrovati pretragu i ispis zapisa na standarni izlaz za jmbg unesen sa standardnog ulaza.

❸ (**30 bodova**) Neka je definisan tip:

```
typedef struct node {
  struct node *data;
  struct node *next; } NODE;
```

kojim je reprezentovan čvor jednostruko povezane liste. Informacioni sadržaj jednostruko povezane liste je adresa prvog čvora neke druge liste. Svaka lista je u određenom nivou. Tako je prva lista na nultom nivou, liste čija je adresa prvog čvora informacioni sadržaj nulte liste su na prvom nivou i tako dalje. Svaki čvor liste od glave prema kraju liste ima redni broj od 0 pa na više, respektivno. Napisati funkciju čiji je prototip:

```
int obrisi_podliste(NODE **glava, int nivo, int pozicija);
```

Funkcija u prosleđenoj listi briše podlistu čija je glava informacioni sadržaj čvora koji je na datoj poziciji i nivou. Funkcija kao rezultat vraća vrijednost 1 ako je uspješno izvršeno brisanje ili vrijednost 0 ako nije pronađen čvor na datom nivou i poziciji.

❹ (**20 bodova**) Neka je definisan tip:

```
typedef struct node {
  int *data; // dinamički podatak
  struct node *left, *right;
} NODE;
```

kojim se reprezentuje čvor binarnog stabla pretrage.

Neka je definisan tip:

```
typedef struct graf
{
  int n;            // broj cvorova
  int *info;  // inf.sadrzaj cvorova
  int **ms; // matrica susjednosti
} GRAF;
```

kojim se reprezentuje matrična reprezentacija neusmjerenog grafa u kojem se informacioni sadržaj čvorova i matrica susjednosti dinamički alociraju.

Napisati program u kojem je potrebno kreirati binarno stablo pretrage (unosom podataka sa standardnog ulaza) te isto nakon toga konvertovati u prethodno definisani tip graf.