

---

PROGRAMIRANJE II (04.07.2017.)

---

- ① (20 bodova) Napisati funkciju koja ima sljedeći prototip:

```
char* spoji(char* (*f)(char*, char*), int n, ...);
```

Funkcija kao argumente prima pokazivač na funkciju koja vrši konkatenciju dva stringa, broj stringova  $n$ , te stringove kao neobavezne argumente, respektivno. Funkcija kao rezultat vraća string koji je nastao konkatencijom neobaveznih argumenata i upotrebom funkcije za konkatenciju koja je data kao argument.

- ② (20 bodova) Data je binarna datoteka DOUBLE.DAT nepoznate dužine, u koju su redom upisani podaci tipa **double**. Napisati program koji treba da sortira podatke u toj datoteci (rezultat izvršavanja programa je sortirana datoteka DOUBLE.DAT). Sortiranje je neophodno realizovati bez učitavanja datoteke u memoriju i korišćenja pomoćne datoteke, korištenjem **insertion-sort** algoritma. U glavnom programu, demonstrirati binarnu pretragu sortirane datoteke (ključ pretrage se navodi kao argument komandne linije), pri čemu je definisana funkcija za pretragu sa sljedećim prototipom:

```
int bin_search(double *niz, int n, double *kljuc);
```

- ③ (30 bodova) Neka su date sljedeće strukture:

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

typedef struct queue {
    int front, rear;
    int size;
    NODE **array;
} QUEUE;
```

koje predstavljaju čvor stabla i red, respektivno. Definisane su funkcije za dodavanje i brisanje elementa iz reda:

```
void enqueue(NODE *root, QUEUE *queue);
NODE* dequeue(QUEUE *queue);
```

Napisati funkciju koja, koristeći red, dodaje novi čvor u skoro kompletno stablo i koja ima sljedeći prototip:

```
void insert(NODE **root, int data, QUEUE* queue);
```

Sve dodatne pomoćne funkcije, u slučaju korištenja, potrebno je definisati.

- ④ (30 bodova) Neka je data sekvencijalna reprezentacija grafa. Informacioni sadržaj grafa je pokazivač na glavu jednostruko povezane liste. Informacioni sadržaj liste je karakter. Napisati funkciju koji omogućava obilazak grafa po dubini pri čemu se za svaki posjećeni čvor grafa ispisuje sadržaj liste. U glavnom programu kreirati liste koje pripadaju grafu i graf te pozvati funkciju za obilazak grafa.

---

PROGRAMIRANJE II (04.07.2017.)

---

- ① (20 bodova) Napisati funkciju koja ima sljedeći prototip:

```
char* spoji(char* (*f)(char*, char*), int n, ...);
```

Funkcija kao argumente prima pokazivač na funkciju koja vrši konkatenciju dva stringa, broj stringova  $n$ , te stringove kao neobavezne argumente, respektivno. Funkcija kao rezultat vraća string koji je nastao konkatencijom neobaveznih argumenata i upotrebom funkcije za konkatenciju koja je data kao argument.

- ② (20 bodova) Data je binarna datoteka DOUBLE.DAT nepoznate dužine, u koju su redom upisani podaci tipa **double**. Napisati program koji treba da sortira podatke u toj datoteci (rezultat izvršavanja programa je sortirana datoteka DOUBLE.DAT). Sortiranje je neophodno realizovati bez učitavanja datoteke u memoriju i korišćenja pomoćne datoteke, korištenjem **insertion-sort** algoritma. U glavnom programu, demonstrirati binarnu pretragu sortirane datoteke (ključ pretrage se navodi kao argument komandne linije), pri čemu je definisana funkcija za pretragu sa sljedećim prototipom:

```
int bin_search(double *niz, int n, double *kljuc);
```

- ③ (30 bodova) Neka su date sljedeće strukture:

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

typedef struct queue {
    int front, rear;
    int size;
    NODE **array;
} QUEUE;
```

koje predstavljaju čvor stabla i red, respektivno. Definisane su funkcije za dodavanje i brisanje elementa iz reda:

```
void enqueue(NODE *root, QUEUE *queue);
NODE* dequeue(QUEUE *queue);
```

Napisati funkciju koja, koristeći red, dodaje novi čvor u skoro kompletno stablo i koja ima sljedeći prototip:

```
void insert(NODE **root, int data, QUEUE* queue);
```

Sve dodatne pomoćne funkcije, u slučaju korištenja, potrebno je definisati.

- ④ (30 bodova) Neka je data sekvencijalna reprezentacija grafa. Informacioni sadržaj grafa je pokazivač na glavu jednostruko povezane liste. Informacioni sadržaj liste je karakter. Napisati funkciju koji omogućava obilazak grafa po dubini pri čemu se za svaki posjećeni čvor grafa ispisuje sadržaj liste. U glavnom programu kreirati liste koje pripadaju grafu i graf te pozvati funkciju za obilazak grafa.