

## PROGRAMIRANJE II (10.10.2018.)

- ① (25 bodova) Napisati funkciju čiji je prototip:

```
void create(unsigned char ***niz, int
**brojBajtova, int n, ...);
```

Funkcija kao neobavezne argumente prima pokazivače na otvorene binarne datoteke te kopira sadržaj u redove dvodimenzionog niza (promjenljiva *niz*). Promjenljiva *n* predstavlja broj pokazivača na otvorene binarne datoteke koji su neobavezni argumenti. Promjenljiva *brojBajtova* predstavlja dinamički alociran (u funkciji) jednodimenzioni niz u kojem svaki član niza predstavlja broj bajtova jedne od otvorenih datoteka. Promjenljiva *niz* predstavlja dinamički alociran (u funkciji) niz nizova u kojem svaki od redova predstavlja sadržaj jedne od prosljeđenih binarnih datoteka.

- ② (25 bodova) Napisati funkciju koja *merge-sort* algoritmom vrši indirektno sortiranje polazne kolekcije bajtova (funkcija ne vrši promjenu rasporeda u datoj polaznoj kolekciji, nego se sortira odgovarajuća kolekcija pokazivača). Prototip funkcije je:

```
void msort (char **a, int begin, int end, int
(*cmp)(const char *, const char *));
```

pri čemu je *cmp* - funkcija za poređenje dva elementa polazne kolekcije koju je potrebno implementirati. U glavnom programu definisati i inicijalizovati niz koji se sastoji od slova alfabeta. Korištenjem funkcije *msort* sortirati, a zatim ispisati sortirani niz slova.

- ③ (25 bodova) Neka su definisani tipovi:

```
typedef struct sll_node {
    void *data; // din. inf. sadržaj
    struct sll_node *next;
} SLL_NODE;
typedef struct dll_node {
    void *data; // din. inf. sadržaj
    struct dll_node *left, *right;
} DLL_NODE;
```

kojim se reprezentuju čvor jednostruko ulančane liste i čvor dvostruko ulančane liste, respektivno.

Napisati funkciju koja dodaje novi podatak (*data*) na kraj jednostruko ili dvostruko ulančane liste, pri čemu parametar *head* predstavlja podatak tipa *SLL\_NODE\*\** ili *DLL\_NODE\*\** u zavisnosti od toga da li se podatak dodaje u jednostuko ili dvostruko ulančanu listu. Ako je vrijednost parametra *isDLL=0*, tada se podatak dodaje u jednostruko ulančanu listu; u suprotnom, podatak se dodaje u dvostruko ulančanu listu. Prototip funkcije je:

```
void add(void *head, void *data, int isDLL);
```

Napisati funkciju, korištenjem prethodno definisane funkcije, koja formira i vraća dvostruko ulančanu listu na osnovu jednostruko ulančane liste (kopiranje sadržaja jednostruko ulančane liste u dvostruko ulančanu listu). Kao drugi parametar, funkcija prihvata pokazivač na funkciju koja se koristi za pravljenje (dinamičke) kopije informacionog sadržaja čvora. Prototip funkcije je:

```
DLL_NODE* copy(SLL_NODE *head,
void* (*f)(const void *data));
```

- ④ (25 bodova) Neka je data ulančana reprezentacija grafa pri čemu je sadržaj čvora grafa realan broj. Definirati strukturu *GRAF* te napisati funkciju čiji je prototip:

```
double** bfs_m(GRAF *g, int start);
```

Funkcija treba da (dinamički) kreira i vrati matricu realnih brojeva, pri čemu *i*-tu vrstu matrice čine vrijednosti čvorova grafa koji su susjedi *i*-tom čvoru grafa. Funkcija obilazi graf po širini (BFS), pri čemu *start* predstavlja indeks čvora od kojeg počinje obilazak.

## PROGRAMIRANJE II (10.10.2018.)

- ① (25 bodova) Napisati funkciju čiji je prototip:

```
void create(unsigned char ***niz, int
**brojBajtova, int n, ...);
```

Funkcija kao neobavezne argumente prima pokazivače na otvorene binarne datoteke te kopira sadržaj u redove dvodimenzionog niza (promjenljiva *niz*). Promjenljiva *n* predstavlja broj pokazivača na otvorene binarne datoteke koji su neobavezni argumenti. Promjenljiva *brojBajtova* predstavlja dinamički alociran (u funkciji) jednodimenzioni niz u kojem svaki član niza predstavlja broj bajtova jedne od otvorenih datoteka. Promjenljiva *niz* predstavlja dinamički alociran (u funkciji) niz nizova u kojem svaki od redova predstavlja sadržaj jedne od prosljeđenih binarnih datoteka.

- ② (25 bodova) Napisati funkciju koja *merge-sort* algoritmom vrši indirektno sortiranje polazne kolekcije bajtova (funkcija ne vrši promjenu rasporeda u datoj polaznoj kolekciji, nego se sortira odgovarajuća kolekcija pokazivača). Prototip funkcije je:

```
void msort (char **a, int begin, int end, int
(*cmp)(const char *, const char *));
```

pri čemu je *cmp* - funkcija za poređenje dva elementa polazne kolekcije koju je potrebno implementirati. U glavnom programu definisati i inicijalizovati niz koji se sastoji od slova alfabeta. Korištenjem funkcije *msort* sortirati, a zatim ispisati sortirani niz slova.

- ③ (25 bodova) Neka su definisani tipovi:

```
typedef struct sll_node {
    void *data; // din. inf. sadržaj
    struct sll_node *next;
} SLL_NODE;
typedef struct dll_node {
    void *data; // din. inf. sadržaj
    struct dll_node *left, *right;
} DLL_NODE;
```

kojim se reprezentuju čvor jednostruko ulančane liste i čvor dvostruko ulančane liste, respektivno.

Napisati funkciju koja dodaje novi podatak (*data*) na kraj jednostruko ili dvostruko ulančane liste, pri čemu parametar *head* predstavlja podatak tipa *SLL\_NODE\*\** ili *DLL\_NODE\*\** u zavisnosti od toga da li se podatak dodaje u jednostuko ili dvostruko ulančanu listu. Ako je vrijednost parametra *isDLL=0*, tada se podatak dodaje u jednostruko ulančanu listu; u suprotnom, podatak se dodaje u dvostruko ulančanu listu. Prototip funkcije je:

```
void add(void *head, void *data, int isDLL);
```

Napisati funkciju, korištenjem prethodno definisane funkcije, koja formira i vraća dvostruko ulančanu listu na osnovu jednostruko ulančane liste (kopiranje sadržaja jednostruko ulančane liste u dvostruko ulančanu listu). Kao drugi parametar, funkcija prihvata pokazivač na funkciju koja se koristi za pravljenje (dinamičke) kopije informacionog sadržaja čvora. Prototip funkcije je:

```
DLL_NODE* copy(SLL_NODE *head,
void* (*f)(const void *data));
```

- ④ (25 bodova) Neka je data ulančana reprezentacija grafa pri čemu je sadržaj čvora grafa realan broj. Definirati strukturu *GRAF* te napisati funkciju čiji je prototip:

```
double** bfs_m(GRAF *g, int start);
```

Funkcija treba da (dinamički) kreira i vrati matricu realnih brojeva, pri čemu *i*-tu vrstu matrice čine vrijednosti čvorova grafa koji su susjedi *i*-tom čvoru grafa. Funkcija obilazi graf po širini (BFS), pri čemu *start* predstavlja indeks čvora od kojeg počinje obilazak.