

PROGRAMIRANJE II (29.01.2019.)

- ① (25 bodova) Napisati funkciju koja kao obavezni parametar prihvata broj n , a zatim n stringova (neobavezni parametri) na osnovu kojih formira i vraća dinamički niz dinamičkih stringova. Prototip funkcije je:

```
char** to_array(int n, ...);
```

Napisati funkciju koja *insert-sort* algoritmom sortira (rastuće) niz stringova, a čiji je prototip:

```
void str_sort(int n, char **array);
```

Napisati glavnu funkciju u kojoj stringove "Jabuka", "Slika", "Kutija", "Avion" i "Kruska" treba, korištenjem funkcije *to_array*, pretvoriti u dinamički niz dinamičkih stringova, a zatim dobijeni niz sortirati (korištenjem funkcije *str_sort*) i ispisati na standardni izlaz.

- ② (25 bodova) U ulaznoj binarnoj datoteci, u prva dva bajta (prvo lakši, a potom teži bajt) upisana je informacija o broju podataka koji su upisani u datu datoteku, dok je u sledećem bajtu upisana informacija o veličini jednog upisanog podatka. Nakon toga su upisani podaci.

Napisati funkciju koja podatke iz datoteke učitava u memoriju (*heap*) i vraća adresu početka podataka u memoriji, dok informacije o broju podataka i veličini jednog podatka vraća preko parametara n (broj podataka) i s (veličina jednog podatka).

```
void* read_f(FILE *f, short *n, unsigned char *s);
```

Napisati glavnu funkciju u kojoj treba, korištenjem funkcije *read_f*, pročitati podatke (brojevi u pokretnom zarezu tipa *float*) iz datoteke "FLOATS.DAT", a zatim proveriti (sekvencijanom pretragom) da li se među učitanim podacima nalazi broj 0 (nula).

- ③ (25 bodova) Neka je dat tip:

```
typedef struct node {  
    char *str; // dinamički string  
    struct node *next;  
} NODE;
```

kojim se reprezentuje čvor jednostruko ulančane liste stringova, te tip:

```
typedef struct sll {  
    NODE *head, *tail;  
} SLL;
```

kojim se reprezentuje jednostruko ulančana lista stringova.

Napisati funkciju koja dodaje novi string na kraj liste:

```
void add(SLL *list, const char *str);
```

Napisati funkciju koja formira jednostruko ulančanu listu na osnovu niza stringova, a čiji je prototip:

```
SLL* to_list(char **array, int n);
```

- ④ (25 bodova) Neka je dat tip:

```
typedef struct bst_node {  
    SLL *list;  
    struct bst_node *left, *right;  
} BST_NODE;
```

kojim se reprezentuje čvor stabla binarne pretrage, pri čemu informacioni sadržaj čvora predstavlja lista stringova koji počinju istim (malim) slovom.

Napisati funkciju koja dodaje novi string u stablo binarne pretrage, i to na kraj odgovarajuće liste korištenjem funkcije *add* iz prethodnog zadatka. Ako dati string ne počinje malim slovom, ignorisati pokušaj dodavanja stringa. Prototip funkcije je:

```
BST_NODE* add_str(BST_NODE *root, const char *str);
```

Napisati rekurzivnu funkciju koja vraća listu koja sadrži stringove koji počinju slovom koje se navodi kao parametar funkcije. Prototip funkcije je:

```
SLL* find(BST_NODE *root, char c);
```

PROGRAMIRANJE II (29.01.2019.)

- ① (25 bodova) Napisati funkciju koja kao obavezni parametar prihvata broj n , a zatim n stringova (neobavezni parametri) na osnovu kojih formira i vraća dinamički niz dinamičkih stringova. Prototip funkcije je:

```
char** to_array(int n, ...);
```

Napisati funkciju koja *insert-sort* algoritmom sortira (rastuće) niz stringova, a čiji je prototip:

```
void str_sort(int n, char **array);
```

Napisati glavnu funkciju u kojoj stringove "Jabuka", "Slika", "Kutija", "Avion" i "Kruska" treba, korištenjem funkcije *to_array*, pretvoriti u dinamički niz dinamičkih stringova, a zatim dobijeni niz sortirati (korištenjem funkcije *str_sort*) i ispisati na standardni izlaz.

- ② (25 bodova) U ulaznoj binarnoj datoteci, u prva dva bajta (prvo lakši, a potom teži bajt) upisana je informacija o broju podataka koji su upisani u datu datoteku, dok je u sledećem bajtu upisana informacija o veličini jednog upisanog podatka. Nakon toga su upisani podaci.

Napisati funkciju koja podatke iz datoteke učitava u memoriju (*heap*) i vraća adresu početka podataka u memoriji, dok informacije o broju podataka i veličini jednog podatka vraća preko parametara n (broj podataka) i s (veličina jednog podatka).

```
void* read_f(FILE *f, short *n, unsigned char *s);
```

Napisati glavnu funkciju u kojoj treba, korištenjem funkcije *read_f*, pročitati podatke (brojevi u pokretnom zarezu tipa *float*) iz datoteke "FLOATS.DAT", a zatim proveriti (sekvencijanom pretragom) da li se među učitanim podacima nalazi broj 0 (nula).

- ③ (25 bodova) Neka je dat tip:

```
typedef struct node {  
    char *str; // dinamički string  
    struct node *next;  
} NODE;
```

kojim se reprezentuje čvor jednostruko ulančane liste stringova, te tip:

```
typedef struct sll {  
    NODE *head, *tail;  
} SLL;
```

kojim se reprezentuje jednostruko ulančana lista stringova.

Napisati funkciju koja dodaje novi string na kraj liste:

```
void add(SLL *list, const char *str);
```

Napisati funkciju koja formira jednostruko ulančanu listu na osnovu niza stringova, a čiji je prototip:

```
SLL* to_list(char **array, int n);
```

- ④ (25 bodova) Neka je dat tip:

```
typedef struct bst_node {  
    SLL *list;  
    struct bst_node *left, *right;  
} BST_NODE;
```

kojim se reprezentuje čvor stabla binarne pretrage, pri čemu informacioni sadržaj čvora predstavlja lista stringova koji počinju istim (malim) slovom.

Napisati funkciju koja dodaje novi string u stablo binarne pretrage, i to na kraj odgovarajuće liste korištenjem funkcije *add* iz prethodnog zadatka. Ako dati string ne počinje malim slovom, ignorisati pokušaj dodavanja stringa. Prototip funkcije je:

```
BST_NODE* add_str(BST_NODE *root, const char *str);
```

Napisati rekurzivnu funkciju koja vraća listu koja sadrži stringove koji počinju slovom koje se navodi kao parametar funkcije. Prototip funkcije je:

```
SLL* find(BST_NODE *root, char c);
```