

PROGRAMIRANJE II (20.09.2023)

- ❶ (20 bodova) Napisati funkciju koja iz ulazne tekstualne datoteke učitava i vraća sve linije koje zadovoljavaju proizvoljan uslov definisan drugim argumentom funkcije:

```
char** linije(FILE *dat, int (*validiraj)(const char *));
```

Napisati funkciju *procitaj* koja prihvata prirodan broj n i n uređenih parova ($2n$ argumenata), gdje prvi element predstavlja datoteku iz koje se čita, dok drugi element predstavlja string koji ukazuje na uslov koji treba proslijediti prilikom čitanja iz fajla. Ukoliko string ima vrijednost "recenica" potrebno je provjeriti da li linija počinje sa velikim slovom, a završava sa tačkom. Za sve druge vrijednosti stringa potrebno je vratiti sve linije iz fajla koje imaju manje od 100 karaktera. Funkcija *procitaj* treba da vrati sve pročitane linije iz svih fajlova koje zadovoljavaju svoje uslove. Prototip funkcije *procitaj*:

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

- ❷ (20 bodova) Neka je dat tip:

```
typedef struct {  
    int identifikator;  
    char ime[50];  
    char prezime[50];  
    int rezultat;  
} TAKMICAR;
```

kojim se reprezentuju podaci o jednom takmičaru na nekom takmičenju.

Napisati funkciju koja prihvata i sortira niz od n podataka o takmičarima u rastućem redoslijedu prema rezultatu. Sortiranje realizovati *merge-sort* algoritmom.

Napisati funkciju koja prihvata naziv izlazne binarne datoteke (parametar *dat*) i (neuređen) niz od n podataka o takmičarima. Funkcija treba da, korištenjem prethodno definisane funkcije, sortira podatke o takmičarima, a zatim da sortirane podatke upiše u izlaznu binarnu datoteku. Prototip funkcije je:

```
void pisi(const char *dat, TAKMICAR *niz,  
         int n);
```

U ulaznoj binarnoj datoteci upisan je nepoznat broj podataka o takmičarima. Podaci o takmičarima nisu uređeni. Napisati funkciju koja u ulaznoj binarnoj datoteci čiji je naziv parametar *dat*, pronalazi i vraća podatke o takmičaru sa zadatim identifikatorom. Ukoliko u datoteci nema podataka o takmičaru sa zadatim identifikatorom, funkcija treba da vrati *NULL*. Nije dozvoljeno učitavanje kompletne datoteke u memoriju. Prototip funkcije je:

```
TAKMICAR* find(const char *dat, int  
               identifikator);
```

- ❸ (20 bodova) Neka su definisani tipovi kojima se redom reprezentuju podaci o jednom takmičaru na nekom takmičenju i čvor jednostruko ulančane uređene liste takmičara:

```
typedef struct {  
    int identifikator;  
    char ime[50];  
    char prezime[50];  
    int rezultat;  
} TAKMICAR;  
typedef struct cvor {  
    TAKMICAR *t; // takmicar (din. podatak)  
    struct cvor *sljedeci;  
} CVOR;
```

Neka je dat tip:

```
typedef struct takmicenje {  
    char naziv[256];  
    CVOR *takmicari; // jednostruko ulančana  
    lista takmičara na takmicenju  
} TAKMICENJE;
```

kojim se reprezentuju podaci o jednom takmičenju. Poređak takmičara u listi je u rastućem redoslijedu po rezultatu takmičara.

Napisati funkciju koja za zadato takmičenje vraća zbir pojedinačnih rezultata svih takmičara, a čiji je prototip:

```
int ukupno(TAKMICENJE *t);
```

Napisati funkciju koja dodaje novi rezultat nekom takmičaru na takmičenju. Ako je već dodat takmičar u listu (ako na takmičenju već postoji takmičar sa istim identifikatorom) ažurirati njegov rezultat, inače dodati novog takmičara. Prototip funkcije je:

```
void dodaj(TAKMICENJE *takmicenje,  
           TAKMICAR takmicar);
```

Napisati funkciju koja na osnovu zadatog niza od n takmičenja kreira i vraća jednostruko ulančanu listu u kojoj su takmičari uređeni po ukupnom rezultatu na svim takmičenjima. Prototip funkcije je:

```
CVOR *rang_lista(TAKMICENJE *niz, int n);
```

- ❹ (20 bodova) Neka su dati tipovi kojima se redom reprezentuju čvor binarnog stabla i graf.

```
typedef struct cvor {  
    char node;  
    struct cvor *roditelj, *lijevi, *desni;  
} CVOR;  
typedef struct g {  
    int n;  
    char nodes[100];  
    int ms[100][100];  
} GRAF;
```

Napisati funkciju koja konvertuje binarno stablo u graf. Ukoliko u stablu ima više čvorova nego što se može podržati grafom, vratiti vrijednost *NULL*. Prototip funkcije je:

```
GRAF* konvertuj(CVOR *root);
```

U glavnom programu definisati binarno stablo dubine tri, te upotrebom prethodno navedene funkcije konvertovati stablo u graf. Nakon toga ispisati informacioni sadržaj grafa.