

PROGRAMIRANJE II (30.01.2018.)

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

```
void sort(char **s, int n, int (*f)(const char *));
```

pri čemu je: *s* – pokazivač na dinamički niz stringova, *n* – broj stringova u datom nizu, *f* – pokazivač na funkciju koja provjerava da li dati string ispunjava neki uslov i vraća informaciju o tome (0 – ne ispunjava, 1 – ispunjava).

Funkcija *sort* treba da korištenjem *insert-sort* algoritma sortira niz stringova rastuće, ali samo na onim pozicijama na kojima se nalaze stringovi koji ispunjavaju dati uslov. Ostali stringovi treba da ostanu na svojim pozicijama.

U glavnom programu formirati dinamički niz od pet stringova ("Jabuka", "Slika", "Visnja", "Tabla", "Avion"), a zatim (korištenjem funkcije *sort*) sortirati stringove na pozicijama na kojima se nalaze stringovi dužine pet.

- ❷ (25 bodova) U ulaznoj binarnoj datoteci, u prva četiri bajta upisana je informacija o broju podataka koji su upisani u datu datoteku, dok je u sljedećem bajtu upisana informacija o veličini jednog upisanog podatka. Nakon toga su upisani podaci.

Napisati funkciju koja podatke upisane u datoteku učitava u memoriju (*heap*) i vraća adresu početka podataka u memoriji, dok informacije o broju podataka i veličini jednog podatka vraća preko parametara *nmemb* (broj podataka) i *size* (veličina jednog podatka). Prototip funkcije je:

```
void* d_read(FILE *f,
             unsigned *nmemb, unsigned char *size);
```

U glavnom programu (korištenjem funkcije *d_read*) pročitati podatke iz datoteke čiji je naziv prvi argument komandne linije. Pod pretpostavkom da su u ulaznoj datoteci upisani označeni cjelobrojni podaci date veličine, sekvencijalno pretražiti pročitane podatke i ispisati informaciju o tome da li se među učitanim podacima nalazi broj -1.

- ❸ (25 bodova) Definirati tip *S_BUFFER* koji predstavlja kružni bafer bez prepisivanja, pri čemu se u bafer dodaju ulančane reprezentacije steka čiji sadržaj predstavljaju *fp* podaci sa dvostrukom preciznošću. U bafer se može dodati maksimalno 20 stekova. Za dodavanje novog steka u bafer, potrebno je definisati funkciju koja ima sljedeći prototip:

```
int add(S_BUFFER *buffer, S_NODE **tos);
```

pri čemu je *S_NODE* čvor steka. Implementirati funkciju za ispis sadržaja kružnog bafera, pri čemu je potrebno ispisati sadržaj svakog steka koji se nalazi u baferu (prilikom ispisa skinuti sadržaj sa steka). Smatrati da su date funkcije za provjeru da li je bafer prazan/pun, kao i funkcija za skidanje elementa sa steka, respektivno:

```
int isEmpty(S_BUFFER *buffer);
int isFull(S_BUFFER *buffer);
int pop(S_NODE **tos, double *value);
```

- ❹ (25 bodova) Neka su date strukture jednostuko povezane liste i sekvencijalne (matrične) reprezentacije grafa. Adresa grafa je informacioni sadržaj čvora liste. Informacioni sadržaj čvora grafa je karakter.

Napisati funkciju koja kao parametar prima glavnu listu, a kreira dinamički niz stringova. Svaki član niza (string) nastaje konkatencijom karaktera (koji su informacioni sadržaj čvorova grafa) prilikom DFS obilaska jednog grafa.

PROGRAMIRANJE II (30.01.2018.)

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

```
void sort(char **s, int n, int (*f)(const char *));
```

pri čemu je: *s* – pokazivač na dinamički niz stringova, *n* – broj stringova u datom nizu, *f* – pokazivač na funkciju koja provjerava da li dati string ispunjava neki uslov i vraća informaciju o tome (0 – ne ispunjava, 1 – ispunjava).

Funkcija *sort* treba da korištenjem *insert-sort* algoritma sortira niz stringova rastuće, ali samo na onim pozicijama na kojima se nalaze stringovi koji ispunjavaju dati uslov. Ostali stringovi treba da ostanu na svojim pozicijama.

U glavnom programu formirati dinamički niz od pet stringova ("Jabuka", "Slika", "Visnja", "Tabla", "Avion"), a zatim (korištenjem funkcije *sort*) sortirati stringove na pozicijama na kojima se nalaze stringovi dužine pet.

- ❷ (25 bodova) U ulaznoj binarnoj datoteci, u prva četiri bajta upisana je informacija o broju podataka koji su upisani u datu datoteku, dok je u sljedećem bajtu upisana informacija o veličini jednog upisanog podatka. Nakon toga su upisani podaci.

Napisati funkciju koja podatke upisane u datoteku učitava u memoriju (*heap*) i vraća adresu početka podataka u memoriji, dok informacije o broju podataka i veličini jednog podatka vraća preko parametara *nmemb* (broj podataka) i *size* (veličina jednog podatka). Prototip funkcije je:

```
void* d_read(FILE *f,
             unsigned *nmemb, unsigned char *size);
```

U glavnom programu (korištenjem funkcije *d_read*) pročitati podatke iz datoteke čiji je naziv prvi argument komandne linije. Pod pretpostavkom da su u ulaznoj datoteci upisani označeni cjelobrojni podaci date veličine, sekvencijalno pretražiti pročitane podatke i ispisati informaciju o tome da li se među učitanim podacima nalazi broj -1.

- ❸ (25 bodova) Definirati tip *S_BUFFER* koji predstavlja kružni bafer bez prepisivanja, pri čemu se u bafer dodaju ulančane reprezentacije steka čiji sadržaj predstavljaju *fp* podaci sa dvostrukom preciznošću. U bafer se može dodati maksimalno 20 stekova. Za dodavanje novog steka u bafer, potrebno je definisati funkciju koja ima sljedeći prototip:

```
int add(S_BUFFER *buffer, S_NODE **tos);
```

pri čemu je *S_NODE* čvor steka. Implementirati funkciju za ispis sadržaja kružnog bafera, pri čemu je potrebno ispisati sadržaj svakog steka koji se nalazi u baferu (prilikom ispisa skinuti sadržaj sa steka). Smatrati da su date funkcije za provjeru da li je bafer prazan/pun, kao i funkcija za skidanje elementa sa steka, respektivno:

```
int isEmpty(S_BUFFER *buffer);
int isFull(S_BUFFER *buffer);
int pop(S_NODE **tos, double *value);
```

- ❹ (25 bodova) Neka su date strukture jednostuko povezane liste i sekvencijalne (matrične) reprezentacije grafa. Adresa grafa je informacioni sadržaj čvora liste. Informacioni sadržaj čvora grafa je karakter.

Napisati funkciju koja kao parametar prima glavnu listu, a kreira dinamički niz stringova. Svaki član niza (string) nastaje konkatencijom karaktera (koji su informacioni sadržaj čvorova grafa) prilikom DFS obilaska jednog grafa.