ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMIRANJE II
(21.06.2023)

❶ **(20 bodova**) Napisati funkciju *formiraj* sa promjenljivim brojem argumenata, koja prima naziv tekstualne datoteke *dat*, cijeli broj *n* i *n* pokazivača na funkcije koje primaju naziv tekstualne datoteke, a kreiraju i vraćaju (dinamički) string ili NULL, te kreira i vraća dinamički niz dinamičkih stringova, pri čemu je *i*-ti element niza rezultat poziva funkcije preko *i*-tog pokazivača na funkciju. Prototip funkcije je:

char formiraj(const char *dat, int n, ...);**

Napisati odgovarajuću funkciju *najduza* koja u tekstualnoj datoteci čiji je naziv zadat, pronalazi i vraća dinamički string koji predstavlja najdužu riječ. Ukoliko u datoteci nema riječi, funkcija treba da vrati prazan string. Ukoliko se desi greška pri otvaranju datoteke, funkcija treba da vrati NULL.

U glavnom programu ilustrovati korištenje funkcije *formiraj* tako da se u tekstualnoj datoteci, čiji je naziv prvi argument komandne linije, pronađe i ispiše na standardni izlaz najduža riječ.

Napomena: riječ je jedno ili više uzastopnih znakova, a od druge riječi je odvojena nekom bjelinom. Pretpostaviti da je maksimalna dužina riječi 100 znakova.

❷ **(20 bodova**) Neka je dat tip:

```
typedef struct vozilo {
        char broj_sasije[8];
        char proizvodjac[21];
        char model[21];
        double snaga;
} VOZILO;
```

kojim se reprezentuje tip VOZILO.

```
typedef struct indeks {
        char broj_sasije[8];
        int adresa;
}INDEKS;
```

kojim se reprezentuju indeksni zapisi o adresi početka zapisa vozila u binarnoj datoteci.

Napisati program u kojem treba u binarnoj datoteci, čiji je naziv prvi argument linije, pronaći podatke o vozilu čiji se broj šasije unosi kao drugi argument linije.

Potrebno je iz binarne datoteke, čiji je naziv dat kao treći argument komandne linije, pročitati sve indeksne zapise i formirati dinamički niz u kojem će se, na osnovu unesenog broja šasije (parametar *broj_sasije*), pronaći odgovarajuća adresa početka zapisa o vozilu u datoteci, te na taj način pročitati i ispisati podatke o pronađenom vozilu.

Za pretragu indeksne datoteke koristiti interpolaciono pretraživanje.

❸ **(20 bodova)** Neka je dat tip kojim se reprezentuje kružni bafer sa prepisivanjem:

```
typedef struct red {
        int niz[100]; int f, r;
} RED;
```

Neka je dat tip kojim se reprezentuje čvor jedostruko ulančane liste:

```
typedef struct cvor {
        RED *red;
        struct cvor *next;
} CVOR;
```

Napisati funkciju za dodavanje elementa u kružni bafer.

Napisati funkciju za dodavanje novog bafera na početak jednostruko ulančane liste. Napisati funkciju koja prolazi kroz sve čvorove liste te za svaki čvor ispisuje sadržaj kružnog bafera:

void ispisi(CVOR* glava);

U glavnom programu kreirati jednu listu i tri bafera sa po dva elementa u svakom. Date kružne bafere potrebno je dodati u listu i zatim ilustrovati upotrebu prethodno definisane funkcije.

❹ **(20 bodova)** Neka je dat tip kojim se reprezentuju podaci o procesu:

```
typedef struct proces{
        int id; // identifikator
        char naziv[24];
        int trajanje;
} PROCES;
```

Neka je dat tip:

```
typedef struct s_cvor{
        int id; // identifikator
        int adresa;
        struct s_cvor *l, *d;
} S_CVOR;
```

kojim se reprezentuje čvor stabla binarne pretrage, koje predstavlja *look-up* tabelu. Pomoću *look-up* tabele moguće je pretraživati ulaznu binarnu datoteku u koju su upisani podaci o procesima. Ključ pretrage je identifikator (*id*) procesa. Svaki čvor stabla binarne pretrage, pored ključa pretrage, sadrži i adresu podataka o procesu sa datim identifikatorom, a ta adresa je, zapravo, pozicija na kojoj počinju podaci o procesu sa datim identifikatorom u ulaznoj binarnoj datoteci.

Napisati rekurzivnu funkciju koja dodaje novi čvor u stablo binarne pretrage, odnosno u *look-up* tabelu. Funkcija prihvata (parametar *root*) i vraća korijen stabla. Prototip funkcije je:

S_CVOR* dodaj(S_CVOR *root, int id, int adresa);

Napisati nerekurzivnu funkciju koja omogućava pretragu podataka o procesima koji su upisani u ulaznu binarnu datoteku, čiji je naziv treći parametar funkcije. Za pretragu treba koristiti stablo binarne pretrage (*look-up* tabelu), čiji je korijen prvi parametar funkcije. Ako u datoteci postoje podaci o procesu sa zadatim identifikatorom (drugi parametar funkcije), funkcija treba da učita i vrati podatke o traženom procesu. U suprotnom, funkcija treba da vrati NULL. Prototip funkcije je:

PROCES* pretrazi(S_CVOR *root, int id, const char *naziv_d);