ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMIRANJE II
(23.06.2021)

❶ (**10 bodova**) Napisati funkciju sa promjenljivim brojem argumenata koja prima string *str*, cijeli broj *n* te *n* pokazivača na funkcije koje vrše neke transformacije nad zadatim stringom. Funkcija treba da nad stringom *str* izvrši sve transformacije. Prototip funkcije je:

void fun(char *str, int n, ...);

Napisati glavni program u kojem treba iz ulazne tekstualne datoteke čiji je naziv prvi argument komandne linije, čitati liniju po liniju (pretpostaviti da linija nije duža od 100 karaktera) te nad svakom linijom primijeniti transformacije *trim* i *capitalize* korištenjem funkcije *fun*. Transformisane stringove upisati u izlaznu tekstualnu datoteku čiji je naziv drugi argument komandne linije. Svaki string je potrebno upisati u novom redu. Pretpostaviti da funkcije *trim* i *capitalize* postoje i da su njihovi prototipovi:

void trim(char *);
void capitalize(char *);

❷ (**10 bodova**) Napisati funkciju koja, na osnovu podataka iz binarne datoteke, kreira i kao rezultat vraća dinamički niz cjelobrojnih označenih podataka koji su veći od parametra *k*. Podaci u datoteci su upisani u rastućem redoslijedu. Broj elemenata niza koji se vraća treba se sačuvati putem parametra *p*. U sklopu pretrage prvog člana u datoteci koji ispunjava dati uslov (veći od parametra *k*) potrebno je koristiti eksponencijalnu pretragu. Pri tome nije dozvoljeno učitavati čitav niz u memoriju do trenutka pronalaska prvog segmenta eskponencijalne pretrage u kojem je bar jedan element koji je veći od parametra *k*. Prototip funkcije je:

int* get(char *file, int k, int *p);

U glavnom programu potrebno je ilustrovati upotrebu funkcije *get*. Pri tome prvi dodatni argument komandne linije je naziv datoteke, drugi dodatni argument je parametar *k*, a svaki naredni cjelobrojna označena vrijednost koju je potrebno dodati na kraj niza koji vraća funkcija *get*. Nakon toga potrebno je sortirati novokreirani niz koristeći *bubble-sort* algoritam i ispisati elemente na standardni izlaz.

❸ **(10 bodova)** Neka je dat tip:

```
typedef struct node {
    int info;
    struct node *next;
} NODE;
```

kojim se reprezentuje čvor jednostruko povezane liste.

Napisati funkciju koja dodaje čvor na kraj liste pri čemu je informacioni sadržaj cijeli broj sa steka koji ispunjava odgovarajući uslov. Uslov je ispunjen ukoliko pokazivač na funkciju (parametar *criteria*) koji je proslijeđen kao parametar funkcije vraća vrijednost 1. Smatrati da se koristi ulančana reprezentacija steka i implementirati funkciju *pop* za uklanjanje odgovarajućeg elementa. U glavnom programu ilustrovati korištenje funkcija.

Prototip funkcije za dodavanje je:

void add(NODE head, STACK** tos, int (*criteria)(int number))**

❹ **(10 bodova)** Neka je dat tip:

```
typedef struct d_node {
    char *key, *translation; // din. stringovi
    struct d_node *left, *right;
} D_NODE;
```

kojim se reprezentuje čvor stabla binarne pretrage (rječnik). Pri tome, polje *key* (ključ pretrage) predstavlja riječ na jednom jeziku, a polje *translation* predstavlja prevod ključa na drugi jezik.

Napisati rekurzivnu funkciju koja u rječnik dodaje novi par riječi. Ako ključ postoji u rječniku, tada je potrebno ažurirati prevod ključa. Funkcija treba da vrati korijen stabla. Prototip funkcije je:

D_NODE* put(D_NODE *root, const char *key, const char *translation);

Napisati nerekurzivnu funkciju koja pronalazi i vraća prevod date riječi (ključa). Ako data riječ ne postoji u rječniku, funkcija treba da vrati NULL. Prototip funkcije je:

char* translate(D_NODE *root, const char *key);