

## PROGRAMIRANJE II (23.01.2024)

- ① (20 bodova) Neka je definisan tip:

```
typedef struct student {  
    char *ime, *prezime;  
    double prosjek;  
} STUDENT;
```

kojim se reprezentuje student.

Napisati funkciju čiji je prototip:

```
double calc(double (*f)(STUDENT), int n, ...);
```

Funkcija kao neobavezne argumente prima podatke tipa *STUDENT*, pri čemu *n* predstavlja broj argumenata tog tipa. Funkcija treba da nad svakim neobaveznim argumentom primijeni transformaciju funkcijom *f*. Funkcija vraća aritmetičku sredinu svih transformisanih vrijednosti funkcijom *f*.

U glavnom programu demonstrirati upotrebu funkcije na dva načina, pri čemu je potrebno kreirati minimalno tri neobavezna argumenta, te nakon poziva funkcije ispisati dobijenu vrijednost. U prvom slučaju funkcija za transformaciju vraća prosjek studenta, a u drugom slučaju ukupan broj karaktera imena i prezimena studenta.

- ② (20 bodova) Neka je dat tip:

```
typedef struct {  
    char ime[50];  
    char prezime[50];  
    double prosjek_poena;  
    double prosjek_asistencija;  
} KOSARKAS;
```

kojim se reprezentuju podaci o jednom košarkašu.

Napisati funkciju koja prima binarnu datoteku, te sortira košarkaše u datoj datoteci upotrebom *insertion sort* algoritma u opadajućem redoslijedu na osnovu prosjeka poena. Ukoliko dva igrača imaju jednak prosječni broj poena, koristiti prosječni broj asistencija za uslov sortiranja. Nije dozvoljeno učitavati cijeli niz u memoriju. Potrebno je sortirati košarkaše u mjestu (modifikovati postojeću binarnu datoteku). Prototip funkcije je sljedeći:

```
void sortiraj(FILE* dat);
```

- ③ (20 bodova) Neka su dati tipovi:

```
typedef struct {  
    char name[64];  
    int duration;  
} PROCESS;  
  
typedef struct node {  
    PROCESS p;  
    struct node *next;  
} NODE;
```

kojim se reprezentuju podaci o jednom procesu čvoru ulančane reprezentacije reda procesa, respektivno. Napisati funkciju koja dodaje novi proces u red, a čiji je prototip:

```
void enq(NODE **head, const char *name, int duration);
```

Napisati funkciju koja briše proces iz reda. Funkcija treba da vrati dinamičku kopiju obrisano procesa. Ako je red prazan, funkcija treba da vrati *NULL*. Prototip funkcije je:

```
PROCESS* deq(NODE **head);
```

Napisati funkciju (korištenjem funkcije *deq*) koja izvršava i briše sve procese iz datog reda. Izvršavanje procesa podrazumijeva ispis naziva i trajanja procesa na standardni izlaz. Prototip funkcije je:

```
void execute(NODE **head);
```

- ④ (20 bodova) Neka je definisan tip:

```
typedef struct cvor {  
    char *str; // dinamički string  
    int br; // broj ponavljanja  
    struct cvor *lijevi, *desni;  
} CVOR;
```

kojim se reprezentuje čvor stabla binarne pretrage (*BST*) podataka tipa string, te tip:

```
typedef struct bst {  
    CVOR *korijen; // korijen BST-a  
} BST;
```

kojim se reprezentuje *BST*.

Napisati funkciju koja dodaje novi string (*str*) u *BST*. Ukoliko string već postoji u stablu, ne dodavati novi čvor već uvećati broj ponavljanja tog stringa. Prototip funkcije je:

```
void dodaj(BST *bst, char *str);
```

Napisati funkciju koja obilazi stablo u odgovarajućem poretku tako da se ispišu stringovi u rastućem redoslijedu (leksikografski). Za svaki string ispisati broj ponavljanja. Prototip funkcije je:

```
void ispis(BST *bst);
```

Napisati glavni program u kojem se ilustruje dodavanje sljedećih stringova u stablo binarne pretrage i njihov ispis u rastućem redoslijedu: "stablo", "binarne", "pretrage", "stablo".