

**PROGRAMIRANJE II - K1  
(12.05.2021)**

- ❶ (5 bodova) Napisati rekurzivnu funkciju *invert* koja kao parametre prima niz stringova (parametar *niz*) i cijeli broj *n* koji predstavlja broj stringova u nizu.

Funkcija *invert* invertuje svaki string u nizu i vraća na istu poziciju u niz. Invertovani string ne smije da sadrži cifre. Obradu svakog stringa potrebno je vršiti u odvojenom rekurzivnom pozivu.

Prototip funkcije *invert* je:

```
void invert(char **niz, int n);
```

- ❷ (5 bodova) Napisati funkciju *init* koja kao obavezne parametre prihvata pokazivač na funkciju (parametar *u*), adresu nekog cijelog broja (parametar *pn*) i cijeli broj *n*, a zatim *n* realnih brojeva tipa *double* (neobavezni parametri).

Parametar *u* predstavlja pokazivač na funkciju koja provjerava da li je ispunjen neki uslov za dati realan broj (funkcija vraća vrijednost 0 ako uslov nije ispunjen, a vrijednost 1 ako uslov jeste ispunjen).

Funkcija *init* treba da od realnih brojeva (neobavezni parametri) koji ispunjavaju dati uslov, formira i vrati dinamički niz. Preko parametra *pn* funkcija treba da vrati broj elemenata formiranog niza. Ako dati uslov nije ispunjen niti za jedan realan broj, funkcija *init* treba da vrati NULL.

Prototip funkcije *init* je:

```
double* init(int (*u)(double), int *pn, int n, ...);
```

- ❸ (5 bodova) Neka je dat tip:

```
typedef struct {  
    char autor[30], zaduzeni[30], status[15];  
    int prioritet;  
} ZADATAK;
```

Polja *autor* i *zaduzeni* predstavljaju string u formi *Ime-Prezime*. Polje *prioritet* može imati vrijednosti 0, 1 i 2, pri čemu manja vrijednost predstavlja viši prioritet.

Iz tekstualne datoteke čiji je naziv prvi argument komandne linije, treba učitati formatirano upisane podatke o zadacima (svaki zadatak u novom redu) i u datoteku čiji je naziv drugi argument komandne linije, treba upisati zadatke sortirano opadajuće prema prioritetu. Za sortiranje koristiti *merge sort* algoritam.

Primjer sortiranih podataka u izlaznoj datoteci:

Marko-Markovic	Janko-Jankovic	Aktivan	0
Jovan-Jovanovic	Petar-Petrovic	Testiranje	0
Marko-Markovic	Jovana-Jovic	Aktivan	1
Marko-Markovic	Milana-Milic	Blokiran	2

- ❹ (5 bodova) Napisati nerekurzivnu funkciju *pretrazi* koja binarno pretražuje niz od *n* cijelih brojeva tipa *long long*, pri čemu je ključ pretrage zadat kao parametar funkcije (*kljuc*).

U slučaju uspješne pretrage funkcija *pretrazi* vraća indeks pronađenog elementa, a u slučaju neuspješne pretrage funkcija vraća -1.

Prilikom pretrage, posjećene elemente niza (elemente niza sa kojima je vršeno poređenje) potrebno je upisivati u binarnu datoteku čiji je naziv zadat kao parametar funkcije (parametar *dat\_naziv*).

Prototip funkcije *pretrazi* je:

```
int pretrazi(const char *dat_naziv, long long  
*niz, int n, long long kljuc);
```

**PROGRAMIRANJE II - K1  
(12.05.2021)**

- ❶ (5 bodova) Napisati rekurzivnu funkciju *invert* koja kao parametre prima niz stringova (parametar *niz*) i cijeli broj *n* koji predstavlja broj stringova u nizu.

Funkcija *invert* invertuje svaki string u nizu i vraća na istu poziciju u niz. Invertovani string ne smije da sadrži cifre. Obradu svakog stringa potrebno je vršiti u odvojenom rekurzivnom pozivu.

Prototip funkcije *invert* je:

```
void invert(char **niz, int n);
```

- ❷ (5 bodova) Napisati funkciju *init* koja kao obavezne parametre prihvata pokazivač na funkciju (parametar *u*), adresu nekog cijelog broja (parametar *pn*) i cijeli broj *n*, a zatim *n* realnih brojeva tipa *double* (neobavezni parametri).

Parametar *u* predstavlja pokazivač na funkciju koja provjerava da li je ispunjen neki uslov za dati realan broj (funkcija vraća vrijednost 0 ako uslov nije ispunjen, a vrijednost 1 ako uslov jeste ispunjen).

Funkcija *init* treba da od realnih brojeva (neobavezni parametri) koji ispunjavaju dati uslov, formira i vrati dinamički niz. Preko parametra *pn* funkcija treba da vrati broj elemenata formiranog niza. Ako dati uslov nije ispunjen niti za jedan realan broj, funkcija *init* treba da vrati NULL.

Prototip funkcije *init* je:

```
double* init(int (*u)(double), int *pn, int n, ...);
```

- ❸ (5 bodova) Neka je dat tip:

```
typedef struct {  
    char autor[30], zaduzeni[30], status[15];  
    int prioritet;  
} ZADATAK;
```

Polja *autor* i *zaduzeni* predstavljaju string u formi *Ime-Prezime*. Polje *prioritet* može imati vrijednosti 0, 1 i 2, pri čemu manja vrijednost predstavlja viši prioritet.

Iz tekstualne datoteke čiji je naziv prvi argument komandne linije, treba učitati formatirano upisane podatke o zadacima (svaki zadatak u novom redu) i u datoteku čiji je naziv drugi argument komandne linije, treba upisati zadatke sortirano opadajuće prema prioritetu. Za sortiranje koristiti *merge sort* algoritam.

Primjer sortiranih podataka u izlaznoj datoteci:

Marko-Markovic	Janko-Jankovic	Aktivan	0
Jovan-Jovanovic	Petar-Petrovic	Testiranje	0
Marko-Markovic	Jovana-Jovic	Aktivan	1
Marko-Markovic	Milana-Milic	Blokiran	2

- ❹ (5 bodova) Napisati nerekurzivnu funkciju *pretrazi* koja binarno pretražuje niz od *n* cijelih brojeva tipa *long long*, pri čemu je ključ pretrage zadat kao parametar funkcije (*kljuc*).

U slučaju uspješne pretrage funkcija *pretrazi* vraća indeks pronađenog elementa, a u slučaju neuspješne pretrage funkcija vraća -1.

Prilikom pretrage, posjećene elemente niza (elemente niza sa kojima je vršeno poređenje) potrebno je upisivati u binarnu datoteku čiji je naziv zadat kao parametar funkcije (parametar *dat\_naziv*).

Prototip funkcije *pretrazi* je:

```
int pretrazi(const char *dat_naziv, long long  
*niz, int n, long long kljuc);
```