

Strukture podataka i algoritmi 1

(zadatak - max 30 poena)

Septembar, 2017

Na velikom turniru društvenih igra biće organizovano više mini turnira. Za svaki mini turnir se unosi ime igre i koliko takmičara može da učestvuje. Za istu igru može postojati više mini turnira. Takmičari se prijavljuju za čitav turnir tako što navode svoje ime, za koliko igara se prijavljuju i imena tih igara. Igrači se prijavljuju na mini turnire onim redom kojim se učitavaju, a mini turniri za istu igru se popunjavaju redom kojim su učitani i moguće je da za nekog igrača nema mesta. Smatrati da su imena igrača jedinstvena i da se jedna igrač neće prijaviti više puta za istu igru.

Nakon završenih prijave iz spiska mini turnira uklanjaju se turniri na koje se nije prijavio ni jedan igrač.

Tokom turnira se unose rezultati partija tako što se unese igra, ime prvog igrača, broj poena prvog igrača, ime drugog igrača i njegov broj poena. Najpre treba proveriti ispravnost unetih podataka, tj. da li su igrači prijavljeni za tu igru na istom mini turniru, i ukoliko su rezultata uneti ispravno, ažuriraju se podaci takmičara. Za svakog igrača se za svaku prijavljenu igru vodi evidencija o broju odigranih partija i broj pobeda. U partiji je pobednik igrač koji je osvojio veći broj poena.

Za svaki mini turnir se određuje najbolji igrač, tj. igrač koji ima najbolji prosečan broj poena po odigranoj partiji.

Napisati program koji će iz datoteka učitati podatke o mini turnirima i prijave igrača, zatim rasporediti igrače po mini turnirima i ukloniti iz spiska mini turnire na kojima nema prijavljenih igrača. Nakon toga se unosi ukupan broj odigranih partija na turniru i podaci o svakoj partiji i određuju najbolji igrači na svakom mini turniru. U datoteku *Najbolji.txt* se upisuje za svaki mini turnir najbolji igrač.

(3 poena)

Za rešavanje problema napisati sledeće funkcije:

a) Napisati funkciju **UcitajTurnire** koja iz datoteke učitava i formira niz mini turnira.

(2 poena)

b) Napisati funkciju **UcitajIgrace** koja iz datoteke učitava i formira niz igrača.

(2 poena)

c) Napisati funkciju **Raspodela** koja će takmičare rasporediti po mini turnirima.

(5 poena)

d) Napisati funkciju **ObrisiPrazne** koja iz spiska mini turnira briše turnire bez prijavljenih igrača.

(5 poena)

e) Napisati funkciju **Provera** koja proverava ispravnost unetih rezultata.

(3 poena)

f) Napisati funkciju **UnosRezultata** koja učitava rezultate svih partija i, ukoliko su dobro uneti, ažurira podatke takmičara. Loše uneti rezultati partija se ignorišu.

(4 poena)

g) Napisati funkciju **Najbolji** koja za mini turnir određuje najboljeg igrača.

(4 poena)

h) Napisati funkciju **Ispis** koja u datoteku ispisuje za svaki mini turnir najboljeg igrača.

(2 poena)

Zadatak rešiti bez korišćenja globalnih promenljivih i bez unapred definisanih dužina korišćenih nizova.

Za realizaciju navedenih funkcija dozvoljeno je definisanje pomoćnih funkcija.

1. Definirati strukturu za čuvanje podataka o mini turnirima iz zadatka o turnirima.
2. Definirati funkciju **Najbolji** iz zadatka o turnirima.

3. Šta je rezultat sledećeg koda i zašto?

```
#include<stdio.h>
int main()
{
    int a=1;
    switch(a)
    {
        int b=20;
        case 1: printf("b is %d\n",b);
                break;
        default:printf("b is %d\n",b);
                break;
    }
    return 0;
}
```

4. Na primeru objasniti šta i kako radi sledeća funkcija

```
int fun(unsigned int x)
{
    int count=0;
    while(x)
    {
        count++;
        x = x&(x-1);
    }
    return count;
}
```

5. Data je sledeća definicija

```
enum vrsta {KVADRAT, PRAVOUGAONIK, KRUG};
struct {
    vrsta tip;
    union {
        float a;
        struct {
            float a;
            float b;
        } stranice;
        float r;
    }fig;
} niz[3];
```

Dodeliti vrednosti svakom elementu niza niz, tako da svaki element sadrži informacije o različitim figurama.

6. Dopisati delove koda koji nedostaju tako da se u funkciju **f** učitaju celobrojne vrednosti koje će biti prikazane naredbom **printf** u **main**-u.

```
#include <stdio.h>
main()
{
    int a,*b,**c;

    a=f(b,c);

    printf("%d %d %d\n",a,*b,**c);
}
```

7. Neka je data sledeća definicija

```
char **a;
```

Napisati deo koda koji će učitavati N linija teksta (N se učitava na početku) i sve ih pamti u promeljivoj **a**.

8. Struktura je definisana na sledeći način

```
struct spa{
    char *ime;
    int ocena;};
```

Definisati niz elemenat tipa date strukture i funkciju koja će sa ulaza učitati broj elemenata nizu (unapred nije poznato ograničenje za broj elementa u nizu) i podatke za sve članove niza. Funkcija treba da vrati pokazivač na prvi elemet niza.

9. Ako je cilj napraviti funkciju koja bitovskim operacijama množi broj sa 5 da li će sledeći program raditi korektno i zašto:

```
#include <stdio.h>
int FiveTimes(int a)
{
    int t;
    t = a<<2 + a;
    return t;
}
int main()
{
    int a = 2, b = 3, c = 5;
    printf("%d %d\n",a,FiveTimes(a));
    printf("%d %d\n",b,FiveTimes(b));
    printf("%d %d\n",c,FiveTimes(c));
    return 0;
}
```

10. Šta će biti rezultat rada sledećeg koda

```
#include <stdio.h>
int main()
{
    int i=43;
    printf("%d\n",printf("%d",printf("%d",i)));
    return 0;
}
```

11. Šta je opisano sledećim deklaracijama:

- a. `char* (*f)();`
- b. `void (*f[10]) (int, int);`
- c. `typedef int (*PFI)(char *, char *);`