

Programiranje II

Beleške sa vežbi

Smer *Informatika*

Matematički fakultet, Beograd

Sana Stojanović

21.02.08.

Sadržaj

1 Rešenja nekih zadataka sa ispitnih rokova	3
---	---

1 Rešenja nekih zadataka sa ispitnih rokova

1. Napisati program koji sa standardnog ulaza učitava prvo broj elemenata niza ($n < 50$), zatim elemente niza i određuje i ispisuje na standardni izlaz maksimalni element niza i koliko puta se on pojavio u nizu.

Na primer, ako unesemo:

8

15 2 8 3 13 15 2 15

program treba da ispiše da je maksimalni element niza broj 15 i da se on pojavio u nizu 3 puta.

```
#include <stdio.h>

main()
{
    int a[50];
    int i, n;
    int max, br;

    scanf("%d", &n);

    /* Prvi nacin: pomocu 3 for petlje, jedna za unosenje elemenata niza,
       druga za racunanje maksimuma, treca za brojanje pojava maksimuma */

    //Unosenje elemenata niza
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);

    //Racunanje maksimalnog elementa niza
    max = a[0];
    for(i=1; i<n; i++)
        if (a[i]>max)
            max = a[i];

    //Brojanje pojavljivanja maksimalnog elementa
    br = 0;
    for(i=0; i<n; i++)
        if (a[i]==max)
            br++;

    printf("Maksimalni element je %d i on se pojavio %d puta.\n", max, br);
}
```

```

/* Drugi nacin: 2 for petlje, jedna za unosenje elemenata i odredjivanje
maksimalnog elementa paralelno a druga za brojanje pojava maksimuma

//Unosenje elemenata niza i odredjivanje maksimalnog elementa
scanf("%d", &a[0]);
max = a[0];
for(i=1; i<n; i++)
{
    scanf("%d", &a[i]);
    if (a[i]>max)
        max = a[i];
}

//Brojanje pojavljivanja maksimalnog elementa
br = 0;
for(i=0; i<n; i++)
    if (a[i]==max)
        br++;

*/

/* Treci nacin: pomocu 1 for petlje, sve radimo paralelno

//Unosenje elemenata, odredjivanje maksimalnog elementa i
//broja pojavljivanja
scanf("%d", &a[0]);
max = a[0]; //postavljamo maximum na prvi element i broj pojava
//tekuceg maksimuma je jednak 1
br = 1;

//sada u petlji trazimo novi maksimum
for(i=1; i<n; i++)
{
    scanf("%d", &a[i]);

    //Ako je tekuci element veci od maksimuma on postaje novi
    //maksimum i broj pojava tekuceg maksimuma postavljamo na 1
    if (a[i]>max)
    {
        max = a[i];
        br = 1;
    }
    //a ako je tekuci element jednak maksimumu onda uvecavamo
    //broj pojava tekuceg maksimuma
    else if (a[i]==max)

```

```

        br++;
    }
    */

/* Posmatrajuci prethodno resenje mozemo videti da nam koriscenje niza
vise nije potrebno posto u jednom koraku petlje pristupamo samo
jednom elementu niza. Tako dobijamo resenje koje umesto niza koristi
samo jednu pomocnu lokaciju:

scanf("%d", &element);
max = element;
br = 1;

for(i=1; i<n; i++)
{
    scanf("%d", &element);
    if (element>max)
    {
        max = element;
        br = 1;
    }
    else if (element==max)
        br++;
}
*/
}

```

2. Napisati program koji sa standardnog ulaza učitava neoznačen ceo broj n i zatim ispisuje na standardni izlaz sumu bitova binarne reprezentacije tog celog broja.

Na primer, za uneto $n = 5$ program treba da ispiše 2 (pošto je binarni zapis broja 5: 101), za uneto $n = 11$ treba da ispiše 3 (pošto je binarni zapis broja 11: 1011).

```

#include <stdio.h>

main()
{
    unsigned n;
    int suma;

    scanf("%d", &n);

    /* Racunamo sumu bitova prvo pomocu bitovskih operatora: */

```

```

suma = 0;
for(; n; n>>=1)
    suma = suma + n&1;

printf("Suma bitova broja %d je %d\n", n, suma);

/* Drugi nacin, pomocu celobrojnog deljenja sa 2:

    suma = 0;
    while(n)
    {
        suma = suma + n%2;
        n = n/2;
    }

*/
}

```

3. Napisati program koji u što manjem broju koraka računa sledeću sumu:

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n * \frac{x^{2n+1}}{(2n+1)!}$$

Ceo broj n i realan broj x se unose sa standardnog ulaza.

```

#include <stdio.h>

main()
{
    float f, suma, x;
    int i, n;

    scanf("%d", &n);
    scanf("%f", &x);

    /* Pocetne inicijalizacije */
    suma = x;
    f = x;

    for(i=1; i<=n; i++)
    {
        f = -f * x * x / ((2*i+1)*2*i);
        suma = suma + f;
    }

    printf("Suma prvih %d clanova je %f\n", n, suma);
}

```

```
}
```

4. Napisati program koji iz datoteke čije se ime zadaje kao prvi argument komadne linije učitava podatke o studentima. U svakom redu datoteke se nalaze redom: ime studenta (karakterska niska dužine ne veće od 15 karaktera), prezime studenta (karakterska niska dužine ne veće od 20 karaktera), broj indeksa studenta (ceo broj) i prosek ocena (realan broj). Pretpostavka je da se u datoteci nalaze podaci za maksimalno 100 studenata.

Prvo formirati niz struktura studenata koji treba popuniti čitanjem iz datoteke pa onda nakon toga ispisati podatke o studentima na standardni izlaz (čitajući ovaj put podatke iz niza).

```
#include <stdio.h>

typedef struct student
{
    char ime[15];
    char prezime[20];
    int indeks;
    float prosek;
} student;

main(int argc, char* argv[])
{
    FILE* in;
    int i, j;
    student niz[100];

    if((in = fopen(argv[1], "r")) == NULL)
    {
        fprintf(stderr, "Ne mogu da otvorim datoteku %s za citanje.", argv[1]);
        return -1;
    }

    //Ucitavamo podatke o studentima u niz
    i=0;
    while(1)
    {
        fscanf(in, "%s%s%d%f", &niz[i].ime, &niz[i].prezime, &niz[i].indeks,
                &niz[i].prosek);

        i++;
        if (feof(in))
            break;
    }
}
```

```

//Citamo podatke o studentima iz niza i ispisujemo ih na standardni izlaz
for(j=0; j<i; j++)
    printf("%s %s %d %f\n", niz[j].ime, niz[j].prezime, niz[j].indeks,
           niz[j].prosek);

fclose(f);
}

```

5. Napisati program koji učitava ime datoteke iz komandne linije i zatim određuje maksimalan ceo broj koji se pojavljuje u toj datoteci (u rešenju ne koristiti nizove). Pretpostavljamo da datoteka sadrži samo cele brojeve. Ukoliko u komandnoj liniji nije navedeno ime datoteke, poruku o tome ispisati na *stderr*.

```

#include <stdio.h>
#include <stdlib.h>

main(int argc, char *argv[]) {
    FILE *in;
    int broj;    /* Tekuci broj koji citamo u datoteci */
    int max;    /* Tekuci maksimum */

    if (argc <2)
    {
        fprintf(stderr, "\nNije navedeno ime datoteke \n");
        exit(1);
    }

    /* Otvaramo datoteku za citanje */
    in=fopen(argv[1],"r");

    /* Citamo prvi ceo broj koji se nalazi u datoteci */
    fscanf(in, "%d", &max);
    broj=max;

    /* Dokle god ima brojeva u datoteci, citamo ih jedan po
       jedan i poredimo ih sa tekucim maksimumom */
    while(!feof(in))
    {
        /* Ako smo nasli broj koji je veci od trenutnog
           maksimuma, menjamo vrednost maksimuma */
        if (max<broj)
            max=broj;

        fscanf(in, "%d", &broj);
    }
}

```



```

    }

    printf("\nMaksimalan broj u datoteci je %d\n", max);
    fclose(in);
}

```

6. Napisati program koji sa standardnog ulaza učitava dva stringa. Proveriti da li se svi karakteri prvog stringa nalaze u drugom stringu. Na kraju ispisati odgovarajuću poruku (npr. *aka* se nalazi u *bajka*, *baba* se nalazi u *bajka*, *majka* se ne nalazi u *bajka*. Takođe i *aabb* se nalazi u *ab*, dok se *abc* ne nalazi u *abb*.)

```

#include <stdio.h>

main() {
    char a[10], b[10]; /* Stringovi koje poredimo */
    int i, j;          /* Brojaci u petljama */
    int ind;           /* Indikator koji oznacava da li stringovi
                       ispunjavaju trazeni uslov */

    printf("Unesite prvi string: \n");
    scanf("%s", a);

    printf("Unesite drugi string: \n");
    scanf("%s", b);

    /* Postavljamo indikator na 1, kao pretpostavku da je uslov ispunjen */
    ind = 1;

    /* Proveravamo da li stringovi ispunjavaju trazeni uslov
       dokle god ima jos karaktera u stringu a i dokle je uslov ispunjen */
    for(i=0; a[i] && ind; i++)
    {
        /* Postavljamo indikator na 0 - znaci jos uvek nismo nasli
           tekuci karakter stringa a u stringu b */
        ind = 0;

        /* Proveravamo da li se karakter a[i] nalazi u stringu b */
        for(j=0; b[j]; j++)
            if (a[i]==b[j])
            {
                ind = 1; /* Nasli smo tekuci karakter prvog stringa */
                break;  /* i mozemo da predjemo na sledeci */
            }
    }
}

```

```

    /* Nakon izlaska iz petlje vrednost promenljive ind nam oznacava da li
       smo nasli sve karaktere stringa a u stringu b */
    if (ind)
        printf("Svi karakteri stringa %s se nalaze u stringu %s\n", a, b);
    else
        printf("Svi karakteri stringa %s se ne nalaze u stringu %s\n", a, b);
}

```

7. Napisati C program koji iz komandne linije učitava nazive nekoliko datoteka i ispisuje na standardni izlaz, za svaku datoteku posebno, dekadnu cifru koje se najčešće pojavljuje u toj datoteci. Ako ima više cifri sa istim najvećim brojem pojava, ispisati najmanju. U slučaju greške u otvaranju datoteke, prijaviti odgovarajući komentar na *stderr*.

```

#include <stdio.h>

main(int argc, char *argv[]) {
    FILE* in;
    int cifra[10]; /* Niz u kome cuvamo broj pojavljivanja svake od
                   cifara. Npr, cifra[5] oznacava broj pojavljivanja
                   cifre 5 u tekucoj datoteci */
    char c; /* Tekuci karakter ulazne datoteke */
    int i, j; /* Brojaci u ciklusima */
    int max_i; /* Indeks cifre koja ima najvise pojavljivanja */

    /* Za svaku datoteku cije je ime zadato u komandnoj liniji
       odredjujemo cifru koja se pojavljuje najvise puta */
    for(i=1; i<argc; i++)
    {
        /* Otvaramo datoteku za citanje */
        in=fopen(argv[i], "r");
        if (in==NULL)
        {
            fprintf(stderr, "\nGreska pri otkrivanju datoteke: %s\n", argv[i]);
            exit(1);
        }

        /* Inicijalizacija niza brojaca na nule.
           Primetimo da moramo koristiti drugi brojac j (odn. ne smemo
           koristiti i zbog toga sto je on vezan za citanje komandne
           linije */
        for(j=0; j<10; j++)
            cifra[j]=0;

        /* Citamo jedan po jedan znak datoteke i kada naidjemo na
           dekadnu cifru uvecamo njen broj pojavljivanja */

```

```

while((c=fgetc(in))!=EOF)
    if (c>='0' && c<='9')
        /* Kako je c karakter, da bismo dobili ceo broj koji nam
           je potreban kao indeks niza, oduzimamo '0' */
        cifra[c-'0']++;

/* Trazimo cifru sa najvećim brojem pojavljivanja, odnosno
   njen indeks u nizu (posto je to jedno te isto) */
max_i = 0; /* Inicijalizujemo max_i na prvu cifru u nizu */
for(j=0; j<10; j++)
    if (cifra[j] > cifra[max_i])
        max_i = j;

printf("Cifra koja se najviše puta pojavila u datoteci %s je %d\n",
       argv[i], max_i);
}
}

```