

Programiranje II

Beleške sa vežbi

Smer *Informatika*
Matematički fakultet, Beograd

Sana Stojanović

21.02.07.

Sadržaj

- 1 Obnavljanje gradiva iz prvog semestra - rešenja nekih zadataka sa ispitnih rokova 3

1 Obnavljanje gradiva iz prvog semestra - rešenja nekih zadataka sa ispitnih rokova

1. 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);
}
```

2. 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);
    }

```

3. 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);
}
}

```

4. Napisati program koji iz datoteke "matrica.txt" učitava prvo dimenziju kvadratne matrice $n < 10$ a zatim redom elemente matrice. Datoteka može da izgleda npr.

```

3

1 2 3
4 5 6
7 8 9

```

Nakon toga ispisati na standardni izlaz elemente matrice po trakama paralelnim glavnoj dijagonali. Tj. ispisati elemente matrice na sledeći način (ako posmatramo elemente matrice preko njihovih koeficijenata):

```

a[0][2]
a[0][1] a[1][2]
a[0][0] a[1][1] a[2][2]
a[1][0] a[2][1]
a[2][0]

```

Odnosno, za matricu navedenu u ovom primeru izlaz treba da izgleda ovako:

```

3
2 6
1 5 9
4 8
7

```

```

#include <stdio.h>

```

```

main()
{
    int a[10][10];      /* Matrica koju učitavamo */
    int n;              /* Dimenzija kvadratne matrice */
    FILE* in;
    int i, j, d;

    /* Otvaramo datoteku za citanje */
    in=fopen("matrica.txt", "r");
    if (in==NULL)
    {
        fprintf(stderr, "\nGreska pri otkrivanju datoteke: matrica.txt\n");
        exit(1);
    }

    /* Učitavamo dimenziju kvadratne matrice */
    fscanf(in, "%d", &n);

    /* Učitavamo elemente kvadratne matrice */
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            fscanf(in, "%d", &a[i][j]);

    /* Ispis matrice paralelno glavnoj dijagonali podelicemo na dva dela.
       Ispis matrice iznad glavne dijagonale (uključujući i glavnu
       dijagonalu) i ispis matrice ispod glavne dijagonale */

    /* Ispis matrice iznad glavne dijagonale */
    /* Primetimo da su razlike indeksa elemenata matrice u svakoj traci
       jednake. Iz tog razloga stampanje elemenata ce se odvijati u
       petlji ciji ce brojac predstavljati razliku tih indeksa. */
    for(d = n-1; d >= 0; d--)
    {
        /* U okviru jedne trake primetimo da prvi indeks uvek kreće od 0
           i da raste, a da se drugi indeks završava na n-1 uvek. */
        for(i=0; d+i <= n-1; i++)
            printf("%d ", a[i][d+i]);
        printf("\n");
    }

    /* Ispis matrice ispod glavne dijagonale */
    /* Opet vazi da su razlike indeksa u svakoj traci jednake, jedina
       razlika je u tome sto je sada prvi indeks veci od drugog */
    for(d=1; d <= n-1; d++)
    {
        for(i=d; i <= n-1; i++)

```

```
        printf("%d ", a[i][i-d]);  
    printf("\n");  
}
```