

3. MASS INSTRUKCIJE

15. siječnja 2010. godine

1. ZADATAK

Napisati funkciju koja prima polje brojeva i duljinu samoga polja. Polje treba biti tipa char. Funkcija potom treba uzeti prva tri broja u polju i u pozivajući program preko imena funkcije vratiti koliko puta se taj podniz pojavljuje u ulaznom nizu. Ako se učitava manje od tri broja, u pozivajući program vratiti 0. Također napisati i program koji učitava brojeve dok se ne učitava negativan broj. Potom pomoću funkcije na ekran ispisati broj pojavljivanja podniza unutar niza. Ukoliko je učitano premalo brojeva potrebno je ispisati poruku „Premalo brojeva“ i završiti program. Može se pretpostaviti da neće biti učitano više od 100 brojeva. Koristiti simboličke konstante.

Primjer:

Ulaz: 1 2 3 4 5 6 1 2 3 -2 Izlaz: 2

Ulaz: 1 2 3 4 5 6 -5 Izlaz: 1

Ulaz: 1 2 1 2 1 -1 Izlaz: 2

Ulaz: 1 0 k Izlaz: Premalo brojeva

Rješenje:

```
#include <stdio.h>
int brojPodniz (int *polje, int n)
{
    int i, brojac=0, prvi, drugi, treci;
    if (n<3)
        return 0;
    else
    {
        prvi = polje[0];
        drugi = polje[1];
        treci = polje[2];
        for (i=0;i<n-2;i++)
            if (prvi == polje[i] && drugi == polje[i+1] && treci == polje[i+2])
                brojac++;
        return brojac;
    }
}

int main ()
{
    int polje[100],i=0,broj,duljina;
    do
    {
        scanf ("%d",&broj);
        if (broj>=0)
        {
            polje[i]=broj;
            i++;
        }
    }while(broj>=0);
    duljina = i;
    if (brojPodniz > 0)
        printf ("%d", brojPodniz (polje,duljina));
    else
        printf ("Premalo brojeva\n");

    return 0;
}
```

2. ZADATAK

Napisati funkciju koja prima matricu (broj stupaca i broj redaka unijeti putem tipkovnice, može se pretpostaviti da obje dimenzije neće biti veće od 100 i može se pretpostaviti da će unesene vrijednosti biti ispravne), te putem pokazivača u glavni program vraća indeks onoga stupca čija je suma najmanja. Ukoliko dva stupca imaju jednaku sumu, u pozivajući program vratiti stupac sa manjim indeksom. Napisati i glavni program koji će matricu popuniti pozitivnim cijelim brojevima u intervalu [40,50]. Potom ispisati na ekran stupac koji ima najveću sumu uz pomoć funkcije iz prvog dijela zadatka. Obavezno je koristiti simboličke konstante.

Primjer:

```
4 3
50 40 44
44 43 41
45 45 48
46 40 49
50 44 45 46 Press any key to continue . . . _
```

Rješenje:

```
#include <stdio.h>
int vratiStupac (int *mat, int m, int n, int maxstup)
{
    int i, maxSuma = 0, suma, stupac, j;
    for (i=0; i<m; i++)
        maxSuma += *(mat+i*maxstup);
    stupac = 0;
    for (i=1; i<n; i++)
    {
        suma = 0;
        for (j=0; j<m; j++)
            suma += *(mat+j*maxstup+i);
        if (suma > maxSuma)
        {
            maxSuma = suma;
            stupac = i;
        }
    }
    return stupac;
}

int main ()
{
    int matrica [MAX][MAX], m, n, stupac, i, j;
    srand ((unsigned)time(NULL));
    scanf ("%d %d", &m, &n);
    for (i=0; i<m; i++)
    {
        for (j=0; j<n; j++)
        {
            matrica[i][j] = rand ()%(50-40+1)+40;
            printf ("%d ", matrica[i][j]);
        }
        printf ("\n");
    }
    stupac = vratiStupac (&matrica[0][0], m, n, MAX);
    for (i=0; i<m; i++)
        printf ("%d ", matrica[i][stupac]);
    return 0;
}
```

3. ZADATAK

Napisati funkciju koja prima polje znakova, i svaki znak koji se nalazi iza razmaka pretvori u veliko slovo, a sva ostala slova se pretvaraju u mala slova. Napisati i glavni program koji demonstrira rad funkcije.

Primjer:

```
daNas jE SUNcaN daN
danas Je Suncan Dan
Press any key to continue . . . _
```

Rješenje:

```
#include <stdio.h>
#include <ctype.h>

void funkcijaMijenja (char *p){
    int i=0;
    while (*(p+i)!='\0')
    {
        *(p+i)=tolower(*(p+i));
        if (*(p+i-1)==' ')
            *(p+i)=toupper(*(p+i));
        i++;
    }
    return;
}

int main ()
{
    char niz[80+1];
    gets(niz);
    funkcijaMijenja(niz);
    puts(niz);
    return 0;
}
```

4. ZADATAK

Napisati program koji nasumično generira troznamenkaste pozitivne brojeve dok god se svaki od brojeva ne pojavi barem 100 puta. Potom na ekran ispisati koliko se stvarno puta pojavio koji broj (sve brojeve), te koji se broj pojavio najmanje puta a koji najviše. Ukoliko se neki od brojeva pojavio jednak broj puta bilo najmanjem bilo najvećem, potrebno je ispisati onaj koji ima veću vrijednost samoga broja.

Primjer:

```
Broj 978 se pojavio 119 puta
Broj 979 se pojavio 121 puta
Broj 980 se pojavio 125 puta
Broj 981 se pojavio 128 puta
Broj 982 se pojavio 135 puta
Broj 983 se pojavio 148 puta
Broj 984 se pojavio 137 puta
Broj 985 se pojavio 126 puta
Broj 986 se pojavio 110 puta
Broj 987 se pojavio 125 puta
Broj 988 se pojavio 124 puta
Broj 989 se pojavio 121 puta
Broj 990 se pojavio 140 puta
```

Rješenje:

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

int main ()
{
    int polje[900]={0};
    int i, min, max, minBroj, maxBroj, gotovo;
    srand((unsigned)time(NULL));
    while(1){
        polje[rand()%(999-100+1)+100-100]++;
        gotovo = 1;
        for(i=0;i<900;i++){
            if(polje[i]<100)
                gotovo = 0;
            if (gotovo == 1)
                break;
        }
        min = polje[0];
        minBroj = 100;
        max = polje[0];
        maxBroj = 100;
        for(i=0;i<900;i++){
            {
                if(polje[i]<=min)
                {
                    min=polje[i];
                    minBroj = i+100;
                }
                if(polje[i]>=max)
                {
                    max=polje[i];
                    maxBroj = i+100;
                }
            }
        }
        for (i=0;i<900;i++){
            printf ("Broj %d se pojavio %d puta\n",i+100, polje[i]);
        }
        printf ("\nBroj %d se pojavio min puta i to %d\n",minBroj, min);
        printf ("Broj %d se pojavio max puta i to %d\n",maxBroj,max);
        return 0;
    }
}
```

5. ZADATAK

Napisati funkciju čiji je prototip: `void izdvojiIzraze (char *ulaz, char *idn1, char *idn2, char *idn3);` čija je namjena da izdvoji i u odgovarajuće polje (`idn1`, `idn2` ili `idn3`) pohrani izraze koji se nalaze unutar zagrada for petlje Primjer: `for(i=0;i<5;i++).`

Napisati i glavni program kojim se testira izvođenje funkcije. Pomoću `gets` u polje znakova unese se jednostavna for petlja i potom pomoću funkcije iz prvog dijela zadatka odrede izrazi.

Uputa:

Dobra je praksa koristiti gotovu funkcij `strchr`.

Primjer:

```
for(i=0;i<5;i++)
i=0
i<5
i++
Press any key to continue . . . _
```

Rješenje:

```
#include <stdio.h>
#include <string.h>

void izdvojiIzraze (char *ulaz, char *idn1, char *idn2, char *idn3)
{
    char *p, *k;
    int duljina;
    p = strchr (ulaz, '(');
    k = strchr (ulaz, ';');
    duljina = k-p;
    strncpy (idn1, p+1,duljina-1);
    idn1[duljina-1]='\0';
    p = strchr (ulaz, '(');
    k++;
    k=strchr (k+1, ';');
    duljina = k-p;
    strncpy (idn2, p+1,duljina-1);
    idn2[duljina-1]='\0';
    p = strchr (p+1, '(');
    k = strchr (k+1, ')');
    duljina = k-p;
    strncpy (idn3, p+1,duljina-1);
    idn3[duljina-1]='\0';
    return;
}

int main ()
{
    char polje[50+1];
    char idn1[20+1],idn2[20+1],idn3[20+1];
    gets (polje);
    izdvojiIzraze (polje,idn1,idn2,idn3);
    puts (idn1);
    puts (idn2);
    puts (idn3);
    return 0;
}
```

6. ZADATAK

Napisati funkciju koja će vratiti malo ili veliko slovo. Funkcija ima prototip `char vratiSlovo (void);` Napisati i glavni program, koji će testirati rad funkcije iz a dijela zadatka. Korisnik putem tipkovnice unese željenu duljinu polja (može se pretpostaviti da neće biti unešeno više od 50). Nije potrebno provjeravati unešenu vrijednost. Glavni program popunjava polje znakova sa slovima koja mu vraća funkcija `vratiSlovo`. Na kraju je potrebno ispisati niz pomoću funkcije `puts`, te broj velikih i mali slova koji se nalaze u generiranom nizu (koristiti odgovarajuće funkcije iz biblioteke `ctype.h`).

Primjer:

```
10
Velika: 7 Mala: 3
LQhRfgBMKK
Press any key to continue . . . _
```

Rješenje:

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>

char vratiSlovo ()
{
    char x;
    int i;
    i=rand()%(2-1+1)+1;
    if (i==1)
        x=rand()%('Z'-'A'+1)+'A';
    if (i==2)
        x=rand()%('z'-'a'+1)+'a';
    return x;
}

int main ()
{
    char polje[50+1];
    int n,i;
    int mala=0;
    int velika=0;

    srand((unsigned)time(NULL));
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        polje[i]=vratiSlovo();
        if(isupper(polje[i])!=0)
            velika++;
        else
            mala++;
    }
    polje[i] = '\0';
    printf ("Velika: %d Mala: %d\n", velika,mala);
    puts(polje);
    return 0;
}
```

7. ZADATAK

Palindrom je niz od jednog ili više znakova s osobinom da se isto čita is lijeva na desno is desna nalijevo. Pod pojmom riječ ćemo podrazumijevati uzastopni niz znamenki ili slova engleske abecede, bilo malih bilo velikih. Riječ je odvojena od druge riječi s jednim ili više praznih mjesta ili znakovima interpunkcije. Znaci interpunkcije su svi oni znakovi, koji nisu ni slova ni brojke. Prvi redak standardnogizlaza treba sadržavati broj PALINDROMNIH RIJEČ u ulaznom nizu.

Primjer:

Rješenje:

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int isPalindrom(const char *a ){
    int vel = strlen(a) - 1, i;
    if(vel == 0)
        return 1;
    else if(vel < 0)
        return 0;
    for( i = 0 ; i < vel ; ++i ){
        if( tolower(a[ i ]) != tolower(a[ vel-- ])) {
            return 0;
        }
    }
    return 1;
}

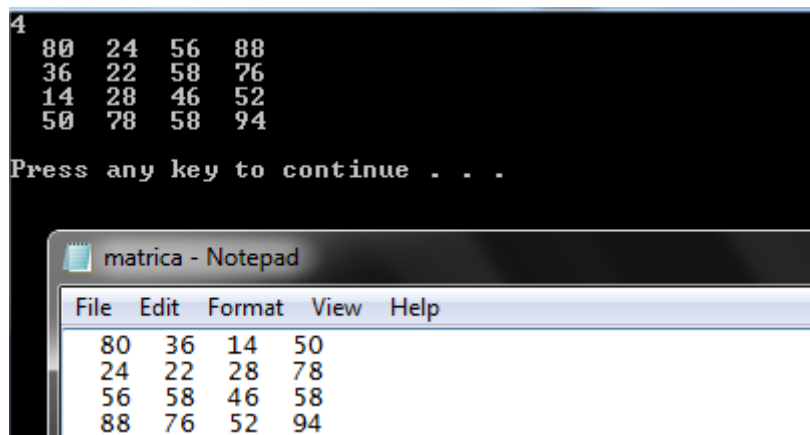
int main (){
    char rec[10000],tmp[10000];
    int i, j, br = 0;
    const char *prazno = "";
    gets(rec);
    for(i = 0, j = 0; i < strlen(rec); ++i){
        if(isalpha(rec[i]) || isdigit(rec[i]))
            tmp[j++] = rec[i];
        else{
            if( isPalindrom( tmp ) ){
                br++;
            }
            memset(tmp, '\0', 1000);
            j = 0;
        }
    }
    if( strcmp(tmp, prazno) != 0 ){
        if( isPalindrom( tmp ) ){
            br++;
        }
    }
    printf( "%d", br );
    return 0;
}
```


8. ZADATAK

Napisati funkciju `zapisiMatricu` koja će zapisati matricu u formatiranu slijednu datoteku i to tako da matrica bude zapisana transponirano. Funkcija prima tok podataka koji će otvoriti glavni program i matricu.

Napisati i glavni program koji će matricu popuniti nasumičnim parnim cijelim dvoznakmenkastim brojevima, (ne većeg ranga od 10 - korisniku ponuditi učitavanje ranga, nije potrebno provjeriti ispravnost unosa), obaviti transponiranje matrice uz pomoć funkcije `transponiraj`, te otvoriti datoteku i uz pomoć funkcije `zapisiMatricu` zapisati i transponirati matricu u datoteku. Potrebno je provjeriti ispravnost otvaranja datoteke. Datoteku nazovite `matrica.txt`. Također u glavnom programu radi provjere ispišite netransponiranu matricu.

Primjer:



The screenshot shows a console window with a black background and white text. It displays a 4x4 matrix of even two-digit numbers. Below the matrix, it says "Press any key to continue . . .". In the foreground, there is a Notepad window titled "matrica - Notepad". The Notepad window shows the transpose of the matrix from the console, which is a 4x4 grid of the same numbers.

```
4
80  24  56  88
36  22  58  76
14  28  46  52
50  78  58  94

Press any key to continue . . .
```

File	Edit	Format	View	Help
80	36	14	50	
24	22	28	78	
56	58	46	58	
88	76	52	94	

Riešenie:

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

void zapisiMatricu (FILE *tok, int *mat, int n, int maxstup)
{
    int t,i,j;
    for (i=0;i<n;i++)
    {
        for (j=0;j<n;j++)
            fprintf (tok, "%4d",*(mat+j*maxstup+i));
        fprintf (tok, "\n");
    }
    return;
}

int main ()
{
    FILE *tokPodataka;
    int matrica[10][10], n, i, j;
    srand ((unsigned) time(NULL));
    tokPodataka = fopen ("c:/matrica.txt","w");
    scanf ("%d",&n);
    for (i=0;i<n;i++)
    {
        for (j=0;j<n;j++)
        {
            matrica[i][j] = (rand()%(49-5+1)+5)*2;
            printf ("%4d", matrica[i][j]);
        }
        printf ("\n");
    }
    zapisiMatricu (tokPodataka, &matrica[0][0], n, 10);
    printf ("\n");
    fclose(tokPodataka);
    return 0;
}
```

9. ZADATAK

U službi za korisnike jednog mobilnog operatera rade vođitelji i studenti. Vođitelji svakog mjeseca dodjeljuju odgovarajuće ocjene na temelju kvalitete rada studenata. Pritom se ocjene upisuju u neformatiranu datoteku „ocjene.dat“ i to šifra studenta (int), 12 ocjena (int) i prosječna mjesečna plaća (double) U drugoj neformatiranoj direktnoj datoteci „studenti.dat“ nalazi se popis svih studenata, sa imenom i prezimenom (char, može se pretpostaviti da zajedno neće biti veći od 20), te šifra studenta. Redni broj u datoteci odgovara šifri studenta. Potrebno je napraviti:

- (1) Program koji će napraviti i popuniti datoteke „ocjene.dat“ i „studenti.dat“ odgovarajućim podacima.
- (2) Program koji će učitati šifru studenta te za datog studenta izračunati prosjek ocjena i na temelju prosjeka dodijeliti odgovarajući bonus. Bonus na kraju godine se računa kao (prosjek ocjena x 100) i izražen je u kunama
- (3) Program koji će učitati šifru te za datog studenta, pročitati i uvećati sve ocjene (ukoliko je ocjena manja od 5) te ih spremiti natrag u datoteku na odgovarajuće mjesto.
- (4) Program koji će učitati realni broj i u formatiranu datoteku „platnaLista.txt“ upisati ime i prezime svih osoba čija je prosječna mjesečna plaća veća od učitane.

```

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

typedef struct {
    int sifra;
    int ocjene[5];
    double placa;
} student_posao;

typedef struct {
    int sifra;
    char ime[20], prezime[20];
} student_privatno;

void punjenje_datoteka(){
    int broj_zapisa, i, j;
    student_posao poslovno;
    student_privatno privatno;
    FILE * studenti, * posao;
    studenti = fopen("studenti.dat", "wb+");
    if (studenti == NULL){
        puts("Greska pri otvaranju datoteke!");
        exit(1);
    }
    posao = fopen("ocjene.dat", "wb+");
    if (posao == NULL){
        puts("Greska pri otvaranju datoteke!");
        exit(1);
    }

    printf("Koliko zapisa zelite?\n");
    scanf("%d", &broj_zapisa);

    for(i = 0; i < broj_zapisa; ++i){
        printf("Unesite poslovne podatke o studentu!\n");
        scanf("%d", &poslovno.sifra);
        for (j = 0; j < 5; ++j){
            scanf("%d", &poslovno.ocjene[j]);
        }
        scanf("%lf", &poslovno.placa);
        fwrite(&poslovno, 1, sizeof(poslovno), posao);

        printf("Unesite privatne podatke o studentu!\n");
        scanf("%d", &privatno.sifra);
        scanf("%s", privatno.ime);
        scanf("%s", privatno.prezime);
        fwrite(&privatno, 1, sizeof(privatno), studenti);
    }

    fclose(posao);
    fclose(studenti);
}

```

```

int main()
{
    int k, sifra;
    float prosjek = 0, bonus;
    double placa;
    student_posao s_posao;
    student_privatno s_privatno;
    FILE * studenti, * posao, * platna;
    studenti = fopen("studenti.dat", "rb+");
    if (studenti == NULL){
        puts("Greska pri otvaranju datoteke!");
        exit(1);
    }
    posao = fopen("ocjene.dat", "rb+");
    if (posao == NULL){
        puts("Greska pri otvaranju datoteke!");
        exit(1);
    }
    /*
    prvi dio zadatka - punjenje datoteke
    */
    /*
    ovo se mora izvršiti prije ostatka zadatka,
    pa zakomentirati i ponovo pokrenuti kod
    jer se pri pozivu fopen s parametrom wb+
    obriše sadržaj datoteke
    */
    punjenje_datoteka();

    /*
    drugi dio zadatka - pretraživanje
    */

    printf("Unesite sifru studenta: ");
    scanf("%d", &sifra);

    while(fread(&s_posao, 1, sizeof(s_posao), posao)){
        if(s_posao.sifra == sifra){
            for(k = 0; k < 5; ++k){
                prosjek += s_posao.ocjene[k];
            }
            prosjek /= 5.;
            break;
        }
    }
    bonus = prosjek * 100;
    printf("\nBonus je %f kn.", bonus);
}

```

```

/*
treći dio zadataka - update datoteke
*/
    fseek (posao,0,SEEK_SET);
    printf("Unesite sifru studenta: ");
    scanf("%d", &sifra);
    while(fread(&s_posao, 1, sizeof(s_posao), posao)){
        if(s_posao.sifra == sifra)
        {
            for(k = 0; k < 5; ++k)
            {
                (if s_posao.ocjene[i]<5)
                    s_posao.ocjene++;
            }
            break
            fseek(posao, -1L*sizeof(s_posao),SEEK_CUR)
            fwrite (&s_posao, sizeof(s_posao), 1, posao);
        }
        fseek (posao,0,SEEK_SET);
    }
/*
četvrti dio zadataka - platna lista
*/
    platna = fopen("platnaLista.txt", "w+");
    if (platna == NULL){
        puts("Greska pri otvaranju datoteke!");
        exit(1);
    }
    printf("\nUnesite placu: ");
    scanf("%lf", &placa);

    while(fread(&s_posao, 1, sizeof(s_posao), posao)){
        if(placa < s_posao.placa){
            while(fread(&s_privatno, 1, sizeof(s_privatno), studenti)){
                if(s_posao.sifra == s_privatno.sifra){
                    fprintf(platna, "%s ", s_privatno.ime);
                    fprintf(platna, "%s\n", s_privatno.prezime);
                }
            }
            fseek(studenti,0L,SEEK_SET);
        }
    }
    fclose(platna);
    fclose(posao);
    fclose(studenti);
    return 0;
}

```