

3. MASOVNE INSTRUKCIJE

17. siječanj 2009.

#1 Zadatak

Putem tipkovnice unesite n, a potom na ekran ispišite n velikih slučajnih slova engleske abecede! Napomena: engleska abeceda ima 26 slova.

Primjer:

10

HCUPPROPAH

Rješenje:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main ()
{
    char slovo;
    int i,n;
    srand(time(NULL));
    scanf ("%d",&n);
    for (i=0;i<n;i++)
    {
        slovo=rand()%('Z'-'A'+1)+'A'; // (90-65+1)+65
        putchar (slovo);
    }
    printf ("\n");
    return 0;
}
```

#2 Zadatak

U glavnom programu unijeti duljinu polja, možete pretpostaviti da polje neće sadržavati više od 100 članova. Funkcija PopuniPolje prima polje i njegovu duljinu te popunjava polje troznamenkastim nasumičnim brojevima i preko imena funkcije vraća broj neparnih brojeva upisanih u polje. Glavni program na ekran potom ispisiuje čitavo polje i broj neparnih brojeva u polju. (Napisati i glavni program i funkciju).

Prototip:

```
int PopuniPolje (int *polje, int duljina)
```

Primjer:

```
10
957 794 164 793 231 994 114 268 673 403
Broj neparnih: 5
```

Rješenje:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX 100
int PopuniPolje (int *polje, int duljina)
{
    int i, brojac=0;
    for (i=0; i<duljina; i++)
    {
        *(polje+i)=rand()%(999-100+1)+100;
        if (*(polje+i)%2==1)
            brojac++;
    }
    return brojac;
}
int main ()
{
    int n, broj, i;
    int polje[MAX];
    srand(time(NULL));
    do
    {
        scanf ("%d", &n);
    } while (n<1 || n>100);
    broj=PopuniPolje(&polje[0], n);
    for (i=0; i<n; i++)
        printf ("%d ", polje[i]);
    printf ("\nBroj neparnih: %d", broj);
    return 0;
}
```

#3 Zadatak

Napišite prototip funkcije koja zbraja sumu matrice!

Rješenje:

```
int suma(int *matrica, int redaka, int stupaca, int maxStupaca);
```

Iako se traži samo prototip evo i same funkcije i odgovarajućeg pozivajućeg programa, zajedno sa zadatkom (nije bilo na masovnicama):

Napisati funkciju koja prima matricu, veličinu matrice i maksimalni broj stupaca kao argumente. Funkcija treba vratiti sumu svih elemenata matrice. Napisati i glavni program koji preko tipkovnice prima dimenzije matrice $m \times n$ (matrica neće biti dimenzija većih od 20×20 – provjeriti!). Matricu potom popuniti nasumičnim brojevima iz intervala unesenog preko tipkovnice. Potom u glavnom programu pozvati funkciju i izračunati sumu matrice. Ispisati matricu i sumu matrice na ekran.

Prototip:

```
int suma(int *matrica, int redaka, int stupaca, int maxStupaca);
```

Primjer:

Unesite dimenzije matrice $m \times n$:

3

4

Unesite donju i gornju granicu brojeva:

-50

50

Matrica:

-18	-21	-14	6
6	-33	-27	-36
29	-39	21	20

Suma: -106

Rješenje:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAXDIM 20
int suma(int *matrica, int redaka, int stupaca, int maxStupaca)
{
    int i,j,rezultat=0;
    for (i=0;i<redaka;i++)
        for (j=0;j<stupaca;j++)
            rezultat+=*(matrica+i*maxStupaca+j);
    return rezultat;
}
int main ()
{
    int i,j,rez,m,n,gornja,donja;
    int matrix[MAXDIM][MAXDIM];
    printf ("Unesite dimenzije matrice m x n:\n");
    do{
        scanf ("%d",&m);
        scanf ("%d",&n);
    }while (m>20||n>20||m<0||n<0);
    printf ("Unesite donju i gornju granicu brojeva\n");
    scanf ("%d",&donja);
    scanf ("%d",&gornja);
    srand(time(NULL));
    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            matrix[i][j]=(rand() % (gornja - donja + 1)) + donja;
    rez=suma(&matrix[0][0],m,n,MAXDIM);
    printf ("\n");
    printf ("Matrica:\n");
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            printf ("%5d",matrix[i][j]);
        printf ("\n");
    }
    printf ("\nSuma: %d\n",rez);
    return 0;
}
```

#4 Zadatak

Napomena:

Moguće da je tekst samog zadatka na masovnim bio drugačiji, ali je ideja identična (ovaj zadatak radim po sjećanju).

Napišite funkciju `ZamijeniRetke` koja će izmijeniti dva slučajno odabrana retka neke matrice. Funkcija prima matricu, broj redaka, broj stupaca i max. stupaca.

Prototip:

```
void ZamijeniRetke(int *matrica, int redaka, int stupaca, int maxStupaca);
```

Primjer:

```
10
Unesite donju i gornju granicu brojeva
0
50

Matrica prije zamjene:
 43  0  12  45  20  21  24  42  3  36
 18 39 38  5  3  49  43 28 50 31
 8  19 24 36 34  3  41  7  40  2
 12 12  6 23 11 25 17 11 30 11
 9  17 16 13 18  4 10  2  3  8
 46 20  8  3  0 39 32 27  7 22
 13  9 33 29 16 17 36 48 12 26
 44 42 36 33 25 16 26 28  0 32

Matrica poslije zamjene:
 43  0  12  45  20  21  24  42  3  36
 18 39 38  5  3  49  43 28 50 31
 8  19 24 36 34  3  41  7  40  2
 12 12  6 23 11 25 17 11 30 11
 9  17 16 13 18  4 10  2  3  8
 44 42 36 33 25 16 26 28  0 32
 13  9 33 29 16 17 36 48 12 26
 46 20  8  3  0 39 32 27  7 22
Press any key to continue . . .
```

Napomena – u rješenje sam dodao i pozivajući program.

Rješenje:

Iako se ne traži, napisao sam i odgovarajući pozivajući program da se možete igrati.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAXDIM 20
void ZamijeniRetke(int *matrica, int redaka, int stupaca, int maxStupaca)
{
    int redak1, redak2, i, t;
    /*generiramo indekse redaka koje želimo zamijeniti*/
    /*naši indeksi su od 0 do redaka-1 pa su takve i granice*/
    redak1=rand()%(redaka-1-0+1)+0;
    do
    {
        redak2=rand()%(redaka-1-0+1)+0;
    }while(redak1==redak2);
    for (i=0;i<stupaca;i++)
    {
        t=matrica[redak1*maxStupaca+i];
        matrica[redak1*maxStupaca+i]=matrica[redak2*maxStupaca+i];
        matrica[redak2*maxStupaca+i]=t;
        /*ALTERNATIVNO:
        t=*(matrica+redak1*maxStupaca+i);
        *(matrica+redak1*maxStupaca+i)=*(matrica+redak2*maxStupaca+i);
        *(matrica+redak2*maxStupaca+i)=t;*/
    }
}
int main ()
{
    int i,j,rez,m,n,gornja,donja;
    int matrix[MAXDIM][MAXDIM];
    printf ("Unesite dimenzije matrice m x n:\n");
    do{
        scanf ("%d",&m);
        scanf ("%d",&n);
    }while (m>20||n>20||m<0||n<0);
    printf ("Unesite donju i gornju granicu brojeva\n");
    scanf ("%d",&donja);
    scanf ("%d",&gornja);
    srand(time(NULL));
    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            matrix[i][j]=(rand() % (gornja - donja + 1)) + donja;
    printf ("\nMatrica prije zamjene:\n");
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            printf ("%5d",matrix[i][j]);
        printf ("\n");
    }
    ZamijeniRetke(&matrix[0][0],m,n,MAXDIM);
    printf ("\nMatrica poslije zamjene:\n");
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            printf ("%5d",matrix[i][j]);
        printf ("\n");
    }
    return 0;
}
```

#5 Zadatak

Napišite funkciju koji će unositi string (ne veći od 30 znakova!) i zamijeniti prvu i drugu polovicu stringa. Ako string ima neparan broj znakova, srednji znak treba ostati na svojem mjestu. Također nakon zamjene lijeve i desne polovice stringa, lijeva polovica treba biti ispisana velikim slovima, a desna polovica ispisana malim slovima. U slučaju neparanog broja znakova, središnji znak nije potrebno dirati!

Napomena – moguće da smo na samom predavanju za ovo velika lijevo, desna malo i drugačije rekli (obrnuto), ali tako i tako je bitna sama mehanika.

```
void PrepolovljenoOkreni(char *p);
```

1# primjer:

ABCdef

DEFabc

2# primjer:

abcdefg

EFGdabc

Rješenje:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
void PrepolovljenoOkreni (char *p)
{
    char t;
    int i,j;
    i=0;
    j=strlen(p)/2;
    if (strlen(p)%2==1)
        j++;
    while (p[j]!='\0')
    {
        t=p[i];
        p[i]=p[j];
        p[j]=t;
        if (islower(p[i]))
            p[i]=toupper(p[i]);
        if (isupper(p[j]))
            p[j]=tolower(p[j]);
        i++;
        j++;
    }
    /*Alternativno:
    while (*(p+j]!='\0')
    {
        t=*(p+i);
        *(p+i)=*(p+j);
        *(p+j)=t;
        if (islower(*(p+i)))
            *(p+i)=toupper(*(p+i));
        if (isupper(*(p+j)))
            *(p+j)=tolower(*(p+j));
        i++;
        j++;
    }*/
    return;
}
int main()
{
    char polje[30+1];
    gets(polje);
    PrepolovljenoOkreni (&polje);
    puts(polje);
    return 0;
}
```


#6 Zadatak

Zadatak preuzet iz ZI2008, kao i službeno rješenje.

Napisati funkciju čiji je prototip:

```
void razdijeli(char *ulaz, char *razdjelnik, char *prvi, char *drugi)
```

Zadani niz razdjelnik dijeli zadani niz ulaz na njegov prvi i drugi dio. Funkcija treba prvi dio niza ulaz prepisati u niz prvi, a drugi dio niza ulaz u niz drugi.

Npr. izvršavanjem naredbi:

```
char prvi[10], drugi[15];
razdijeli("https://ahyco.fer.hr", "://", prvi, drugi);
printf("%s %s", prvi, drugi);
```

na zaslon će se ispisati:

https ahyco.fer.hr

Možete pretpostaviti da niz ulaz uvijek sadrži točno jedan podniz razdjelnik.

Rješenje:

```
void razdijeli(char *ulaz, char *razdjelnik, char *prvi, char *drugi)
{
    char *poz;
    int dulj;
    poz = strstr(ulaz, razdjelnik);
    dulj = poz - ulaz;
    strncpy(prvi, ulaz, dulj);
    prvi[dulj] = 0;
    strcpy(drugi, poz + strlen(razdjelnik));
}
```

Alternativno rješenje:

Pomalo duže rješenje, ali ručno kopiranje. Također pozivajući program možete iskoristiti za pokretanje funkcije iz službenih rješenja.

```
#include <stdio.h>
#include <string.h>
void razdijeli (char *ulaz, char *razdijelnik, char *prvi, char *drugi)
{
    char *p;
    int duljinaPrvog,i;
    p=strstr(ulaz,razdijelnik);
    duljinaPrvog=strlen(ulaz)-strlen(p);
    for(i=0;i<=duljinaPrvog;i++)
        prvi[i]=ulaz[i];
    prvi[i]='\0';
    p=p+strlen(razdijelnik);
    i=0;
    while (p[i]!='\0')
    {
        drugi[i]=p[i];
        i++;
    }
    drugi[i]='\0';
}
int main()
{
    char prvi[10], drugi[15];
    razdijeli("https://ahyco.fer.hr", "://", prvi, drugi);
    printf("%s %s", prvi, drugi);
    return 0;
}
```

#7 Zadatak

Napišite program koji putem tipkovnice prima ime datoteke i n. Program otvara formatiranu datoteku učitano ime i u nju zapisuje tablicu množenja n x n. N mora biti u intervalu [1,15]. Nije potrebno provjeravati jeli tok podataka uspješno otvoren.

Primjer:

Ekran:

Upisite ime datoteke: tablica.txt

10

Datoteka „tablica.txt“:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Napomena – ispravnost ovog zadatka možete provjeriti tako da otvorite folder u koji ste spremili ovaj program. Nakon pokretanja i upisa traženih vrijednosti stvorit će vam se datoteka. Najbolje da ju nazovete tablica.txt, tada ćete ju moći otvoriti u notepadu i provjeriti ispravnost rješenja.

```

#include <stdio.h>
int main()
{
    int n, i, j;
    char imeDat[64+1];
    FILE *tok;
    printf("Upisite ime datoteke: ");
    scanf("%s", imeDat);
    do
    {
        printf("Upisite vrijednost za n: ");
        scanf ("%d",&n);
    } while (n < 1 || n > 15 );
    tok = fopen(imeDat, "w");
    fprintf (tok," ");
    for (i = 1; i <= n; i++)
    {
        fprintf(tok, "%4d", i);
    }
    fprintf(tok, "\n");
    for (i = 1; i <= n; i++)
    {
        fprintf(tok, "%4d", i);
        for (j = 1; j <= n; j++) {
            fprintf(tok, "%4d", i*j);
        }
        fprintf(tok, "\n");
    }
    fprintf(tok, "\n");
    fclose(tok);
    return 0;
}

```

#8 Zadatak

Zadatak preuzet iz ZI2008, kao i službeno rješenje. Za ispravno funkcioniranje potrebne su datoteke koje nisu priložene.

Direktna neformatirana datoteka "artikl.dat" sadrži zapise o artiklima. Svaki zapis sadrži šifru artikla (cijeli broj tipa int), naziv artikla (niz od 50+1 znakova) i cijenu artikla (realni broj tipa double) pri čemu redni broj zapisa u datoteci odgovara šifri artikla. Svaki zapis slijedne formatirane datoteka "narudzba.txt" sadrži šifru artikla (cijeli broj od najviše 6 znamenaka). Napisati program koji će ispisati naziv najjeftinijeg artikla u datoteci "narudzba.txt". Ako postoji više takvih artikala, ispisati bilo kojeg od njih. Možete pretpostaviti da datoteke sadrže barem po jedan zapis i da datoteka "narudzba.txt" sadrži samo artikle iz datoteke "artikl.dat". Nije potrebno provjeravati uspješnost obavljanja operacija nad datotekama.

Rješenje:

```
#include <stdio.h>
#include <string.h>
int main()
{
    FILE *datArtikal, *datNarudzba;
    double min;
    char naziv [50 + 1];
    int sifra;
    struct
    {
        int sifra;
        char naziv[50+1];
        double cijena;
    } artikal;
    datArtikal = fopen ("artikli.dat", "rb");
    datNarudzba = fopen ("narudzba.txt", "r");
    fscanf(datNarudzba, "%d", &sifra);
    fseek(datArtikal, (long)(sifra-1)*sizeof(artikal), SEEK_SET);
    fread(&artikal, sizeof (artikal), 1, datArtikal);
    min = artikal.cijena;
    strcpy(naziv, artikal.naziv);
    while (fscanf(datNarudzba, "%d", &sifra) == 1)
    {
        fseek(datArtikal, (long)(sifra-1)*sizeof(artikal), SEEK_SET);
        fread(&artikal, sizeof (artikal), 1, datArtikal);
        if (artikal.cijena < min)
        {
            min = artikal.cijena;
            strcpy(naziv, artikal.naziv);
        }
    }
    printf ("Najjeftiniji artikal je: %s \n", naziv);
    fclose(datArtikal);
    fclose(datNarudzba);
    return 0;
}
```

#9 Zadatak

U svakom retku slijedne formatirane datoteke nalazi se zapis o studentu koji je oblika:
ssssNNpppppppppppo...

gdje je:

ssss - sifra studenta, 4 znamenke, cijeli broj

NN - broj položenih ispita, cijeli broj

ppp - šifra predmeta, cijeli broj

o – ocjena, cijeli broj

Parova pppo ima onoliko koliko je student položio ispita.

napisati funkciju koja za svakog studenta ispisuje šifru, te prosječnu ocjenu!

Ime datoteke prenosi se u fju kao argument funkcije!

Nije potrebno provjeravati uspješnost operacija!

Rješenje:

```
#include<stdio.h>
void fja(char *ime) {
    FILE *f;
    int sifStud, n, sifPred, ocjena, i;
    double prosOcj;
    f = fopen(ime, "r");
    while(fscanf(f, "%4d%2d", &sifStud, &n) == 2)
    {
        prosOcj = 0;
        for( i=0; i<n; i++)
        {
            fscanf(f, "%3d%1d", &sifPred, &ocjena);
            prosOcj += ocjena;
        }
        prosOcj /= n;
        printf("%d %lf", sifStud, prosOcj);
        fscanf(f, "%*c");
    }
    fclose(f);
    return;
}
```

Dodatna napomena:

Za ispravno funkcioniranje potrebna datoteka koja nije priložena.

Detaljno objašnjenje ovog zadatka imate u Drizztovom tutorialu za funkcije odnosno:

<http://www.fer2.net/showpost.php?p=758420&postcount=2>