

Задачи со решенија за

СТРУКТУРНО ПРОГРАМИРАЊЕ

1. Непарен еквивалент (ФИНКИ - Code - задачи за вежби 2014)

Од тастатура се читаат 2 позитивни цели броеви, поголеми од 0. Доколку се внесе барем еден негативен број или 0 треба да се печати „Invalid input“.

Треба да се определи дали помалиот број “е непарен еквивалент” од поголемиот број.

Еден број “е непарен еквивалент” на друг број, ако и само ако неговите цифри се наоѓаат на непарните позиции од другиот број, во истиот редослед. Соодветно да се испечатат пораки „DA“ и „NE“.

Позициите треба да се сметаат оддесно-налево (најмалку значајната цифра е на позиција 1)

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

int main()
{
    int pogolem, pomal;
    printf("vneste dva broja: ");
    scanf("%d%d",&pogolem,&pomal);
    if(pogolem <=0 || pomal <=0){
        printf("Invalid input");
        return -1;
    }
    int tmp;
    if(pomal>pogolem){
        tmp=pogolem;
        pogolem=pomal;
        pomal=tmp;
    }
    int flag=0;
    while(pomal){
        if(pomal%10!=pogolem%10) {flag =1; printf("NE "); break;}
        pomal=pomal/10;
        pogolem=pogolem/100;
    }
    if(flag==0) printf("DA");
    return 0;
}
```

Задачи за втор колоквиум и испит

- **Задачи со рекурзија**

1. Да се напише програма која за секои два прочитани броја ќе отпечати "da" ако првиот број е степен на вториот, во спротивно да испечати "ne". Проверката дали еден број е степен на друг да се реализира со посебна рекурзивна ф-ција.

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

```
int prover1(int prv,int vtor){
    if (prv==0) return 0;
    if(prv==vtor) return 1;
    else return prover1(prv/vtor,vtor);
}
```

```
int main()
{
    printf("Vnesete dva broja: ");
    int prv,vtor;
    scanf("%d",&prv);
    scanf("%d",&vtor);
    int x=proveri(prv,vtor);
    if(x==1)
        printf("DA");
    else printf("NE");
    return 0;
}
```

2. Од стандарден влез се читаат два цели броеви.

Да се напише програма која ќе отпечати колку пати вториот број се појавува во првиот без преклопување на повторувањата. Ова да се реализира со посебна рекурзивна функција.

Пример:

за парот броеви 5167163 и 16 треба да се отпечати 2.

За парот броеви 34333 и 33 треба да се отпечати 1.

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

```
int broj_povtoruvanje(n1,n2,del){//bez preklopuvanje
    if(n1==0) return 0;
    int pom;
    pom=n1%del;
    if(pom==n2)return 1+broj_povtoruvanje(n1/del,n2,del);
    else return 0+broj_povtoruvanje(n1/del,n2,del);
}
```

```
/*int broj_povtoruvanje(n1,n2,del){//so preklopuvanje
    if(n1==0) return 0;
    int pom;
    pom=n1%del;
    if(pom==n2)return 1+broj_povtoruvanje(n1/10,n2,del);
    else return 0+broj_povtoruvanje(n1/10,n2,del);
}*/
```

```
int main(){
    int n1,n2,tmp;
    printf("Vnesete dva broja: ");
    scanf("%d%d",&n1,&n2);
    if(n2>n1){
        tmp=n1;
```

```

    n1=n2;
    n2=tmp;
}
int pom=n2;
int del=1;
while(pom!=0){
    del*=10;
    pom=pom/10;
}
int n;
n=broj_povtoruvanje(n1,n2,del);
printf("Brojot %d se povtoruva vo brojot %d %d pati.",n2,n1,n);
return 0;
}

```

3. Да се напише рекурзивна функција која ќе ги изброи парните и непарните цифри на даден број. Да се напише и главна програма за тестирање на програмата.

```

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

void presmetaj_rekurzivno(int broj,int *parni, int *neparni){
    if(broj!=0){
        if(((broj%10)%2)==0) (*parni)++;
        else (*neparni)++;
        presmetaj_rekurzivno(broj/10,parni,neparni);
    }
}

int main()
{
    int parni=0,neparni=0,broj=0;
    printf("Vnesete broj: ");
    scanf("%d",&broj);
    presmetaj_rekurzivno(broj,&parni,&neparni);
    printf("Vo brojot %d ima %d parni i %d neparni cifri.",broj,parni,neparni);
    return 0;
}

```

4. Да се напише рекурзија која за даден цел број ќе формира нов број составен само од парните цифри на парни позиции.(ФЕИТ 09.11.2012 термин 2)

primer : broj=12345
rezultat=24
primer : broj=208558
Rezultat=28

```

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

```

```

int rekurzija_parni(int n){
    n=n/10;
    if(n>=10 && n<=99){
        if (((n%10)%2)==0) return n%10;
        else return 0;
    }
    else if(n>=0 && n<=9) {
        if(n%2==0) return n;
        else return 0;
    }
    else if(((n%10)%2)==0) return (n%10)+10*(rekurzija_parni(n/10));
    else return rekurzija_parni(n/10);
}

int main()
{
    int n;
    scanf("%d",&n);
    int x=rekurzija_parni(n);
    printf("%d",x);
    return 0;
}

```

5. Треба да се напише рекурзија која како аргумент прима два цели броја составени од ист број цифри. Функцијата треба да го врати бројот на различни цифри во двата броја кои се наоѓаат на исти позиции. Да се напише главна програма во која се внесуваат непознат број на парови од цели броеви и се проверува колку од нив имаат повеќе од 2 различни цифри на исти позиции. (ФЕИТ - јануарски испит 07.02.2013 термин 2 група 2)

Пример: броевите 1334 и 1344 имаат 1 различна цифра на иста позиција.

```

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

int razlicni_cifri(int n1,int n2){
    if(n1==0) return 0;
    else if((n1%10)!=(n2%10)) return 1 + razlicni_cifri(n1/10,n2/10);
    else return 0 + razlicni_cifri(n1/10,n2/10);
}

int main(){
    int n1,n2,br=0;
    printf("Vnesuvajte gi parovite na broevi: ");
    while(scanf("%d%d",&n1,&n2)){
        if(razlicni_cifri(n1,n2)>2)br++;
        printf("razlicni se %d cifri\n",razlicni_cifri(n1,n2));
    }
    printf("%d parovi imaat poveke od 2 razlicni cifri na isti pozicii.",br);
    return 0;
}

```

6. Мали цифри

Од стандарден влез се читаат N цели броеви. Да се напише програма која за секој прочитан број ќе ги отпечати на стандарден излез само цифрите помали од 5, како и бројот на цифри кои се отпечатени. Печатењето на цифрите помали од 5 да се реализира со посебна рекурзивна функција. (ФИНКИ – Code – пример испитни задачи 2014)

Пример:

за бројот 8423871 треба да се отпечати 4231 : 4

```
#include <stdio.h>

int pecati(int n) {
    if(n==0) return 0;
    int p=pecati(n/10);
    int c=n%10;
    if(c<5) {
        printf("%d",c);
        return p + 1;
    }
    return p;
}

int main() {
    int n,i,x;
    scanf("%d",&n);
    for(i=0;i<n;++i) {
        scanf("%d",&x);
        int r=pecati(x);
        printf(" : %d\n",r);
    }
    return 0;
}
```

7. „Превртување“ на низа

Да се напише рекурзивна функција `prevti` која како аргумент прима низа од n цели броеви, и по нејзиното завршување првичната низа ја „превртува“ (го менува редоследот на нејзините членови). На пример, низата 1, 4, 7, 5, -22, 9 ја претвора во низа 9, -22, 5, 7, 4, 1. (Дозволено е да се додадат дополнителни аргументи на функцијата!).

Да се направи програма која од тастатура ќе прочита цел број n ($0 < n < 100$), и потоа уште n цели броеви – елементи на низата. Низата прво да се испечати на екран, потоа да се повика функцијата `prevti` за дадената низа и на крајот повторно да се испечати низата на екран. (ФИНКИ – Code – пример испитни задачи 2014)

```

#include <stdio.h>

void prevrti(int* a, int n, int i)
{
    if(n > 2*i){
        int pom = a[i];
        a[i] = a[n-i-1];
        a[n-i-1]=pom;
        prevrti(a, n, i+1);
    }
}

/*
void prevrti(int *a, int n, int k){//nerekurzivno reshenie
    int i;
    for(i=0;i<k;i++){
        int pom = a[i];
        a[i] = a[n-i-1];
        a[n-i-1]=pom;
    }
}

*/

int main()
{
    int niza[100], n, i;
    scanf("%d", &n);
    for(i = 0; i < n; i++){
        scanf("%d", &niza[i]);
    }
    for(i = 0; i < n-1; i++)
        printf("%d ", niza[i]);
    printf("%d\n",niza[i]);
    /*//bez rekurzija
    if(n%2!=0) k=n/2+1;//ako nizta ima neparen broj na elementi(sredniot element ne se menuva)
    if(n%2==0) k=n/2;//ako nizata ima paren broj na elementi
    prevrti(niza,n,k);
    */

    prevrti(niza, n, 0);
    for(i = 0; i < n - 1; i++)

```

```

    printf("%d ", niza[i]);
    printf("%d", niza[i]);
    return 0;
}

```

8. Максимум 3

Од стандарден влез се читаат N тројки цели броеви. Да се напише програма која за секои прочитани три цели броја ќе го отпечати најголемиот. Наоѓањето на најголемиот од три броја да се реализира со посебна рекурзивна функција. (ФИНКИ – Code – Пример испитни задачи 2014)

Пример:

за тројката броеви 8 15 3 треба да се отпечати 15

```

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

int max(int a,int b,int c)
{
    if(a>=b && a>=c) {
        return a;
    }
    if(a<b) {
        return max(b,a,c);
    }
    if(a<c) {
        return max(c,b,a);
    }
}

int main ()
{
    int N,i,x,y,z,maximum;
    scanf("%d", &N);
    for (i=0;i<N;i++){
        scanf("%d%d%d",&x,&y,&z);
        maximum=max(x,y,z);
        printf("%d\n",maximum);
    }
    return 0;
}

```


9. Да се напише рекурзивна функција која за даден цел број ќе провери дали има барем една цифра делива со 3. (ФЕИТ – 09.11.2012 термин 3, група 1)

Пример 1:

Број: 1245

Резултат: 0

Пример 2:

Број: 12345

Резултат: 1

```
#include <stdio.h>
#include <stdlib.h>
int deliva_so_3(int x){
    if(x<=9 && x>=0) {
        if(x%3==0) return 1;
        else return 0;
    }
    if((x%10)%3==0) return 1;
    else return deliva_so_3(x/10);
}
int main()
{
    int n,rezultat;
    scanf("%d",&n);
    rezultat=deliva_so_3(n);
    if(rezultat)
        printf("Ima barem edna cifra deliva so 3.");
    else printf("Nema nitu edna cifra deliva so 3.");
    return 0;
}
```

10. Да се напише рекурзија која како аргумент ќе добие низа од цели броеви и како резултат ќе провери дали збирот на елементите од низата после i-тиот елемент е еднаков на i-тиот елемент. Ако сите елементи го исполнуваат условот , функцијата враќа 1, во спротивно враќа 0. (ФЕИТ – 21.08.2013, термин 1, група 2)

Пример : Niza1=[16,8,4,2,1,1]; Niza2=[256,128,64,32,14,7,7]

За Niza1 враќа 1, за Niza2 враќа 0.

```
#include<stdio.h>

int suma_el(int *a, int n)
{
    int i, zbir=0;
    if(n==0)
        return 1;
    if(n==1)
        zbir=a[0];
    if(n>1)
        for(i=1;i<n;i++){
            zbir+=a[i];
        }
    if(a[0]!=zbir) return 0;
    else return suma_el(a+1,n-1);
}
```

```
int main () {
    int a[100], n, i, k;
    scanf("%d", &n);
    for(i=0;i<n;i++)
        scanf("%d", &a[i]);
    k=suma_el(a,n);
    if(k==1)
        printf("DA");
    else
        printf("NE");

    return 0;
}
```

11. Да се напише рекурзивна функција која како аргумент добива цел број. Функцијата како резултат треба да го врати збирот на најзначајната цифра и втората најзначајна цифра на тој број. Да се напише програма за тестирање на функцијата, во која од два цели броја, внесени од тастатура, на екран ќе го испечати оној кој има поголем збир на овие две цифри. (ФЕИТ – 21.08.2012, група 1)

Пример 1: a=12345 b=5678 На екран: 5678

Пример 2: a=1234 b=30321 На екран: broevite se so ist zbir

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

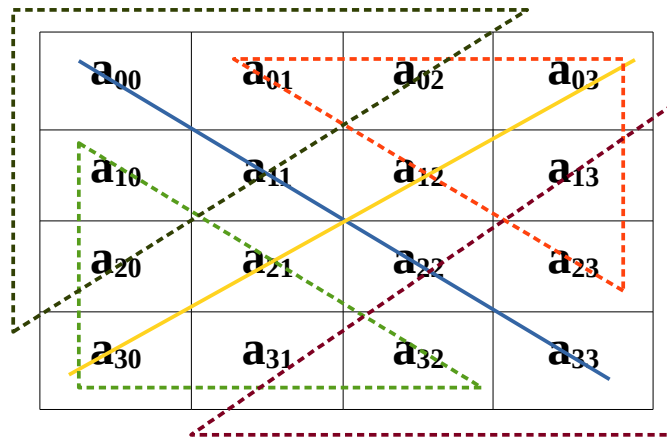
```
#include <stdlib.h>
```

```
int najznacajni_cifri(int broj){  
    if(broj<=99 && broj>=10) return (broj%10 + broj/10);  
    else return najznacajni_cifri(broj/10);  
}
```

```
int main()  
{  
    int n1,n2;  
    scanf("%d%d",&n1,&n2);  
    int max1=najznacajni_cifri(n1);  
    int max2=najznacajni_cifri(n2);  
    if(max1>max2) printf("%d",n1);  
    else if(max2>max1) printf("%d",n2);  
    else printf("broevite se so isti zbir");  
    return 0;  
}
```

- Задачи со матрици

Општ член – a_{ij} со димензии $n \times n$



- Главна дијагонала ($i=j$)
- Споредна дијагонала ($i+j=n-1$)
- - - - - Елементи над главна дијагонала ($i < j$)
- - - - - Елементи под главна дијагонала ($i > j$)
- - - - - Елементи над споредна дијагонала ($i+j < n-1$)
- - - - - Елементи под споредна дијагонала ($i+j > n-1$)

1. Да се напише програма која од тастатура чита две квадратни матрици (најмногу 10×10). Програмата треба да го одреди збирот на квадратот на прво ниво (пример) за двете матрици одделно. На екран да се испечати матрицата која има поголем збир на овие елементи. (ФЕИТ – 21.08.2012 група 1)

Пример :

Прва матрица :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Втора матрица:

4	2	3	4
2	6	7	2
3	10	11	1
5	6	1	6

На екран:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10

int main()
{
    int a[MAX][MAX],b[MAX][MAX];
    int n,m,i,j;

    printf("Vnesete go brojot na redici, pa brojot na koloni na matricata: ");
    scanf("%d%d",&m,&n);
    printf("Vnesete ja matricata:\n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    int zbir=0;
    for(i=0;i<n;i++)
        zbir+=a[0][i];
    for(i=0;i<n;i++)
        zbir+=a[m-1][i];
    for(i=0;i<m;i++)
        zbir+=a[i][0];
    for(i=0;i<m;i++)
        zbir+=a[i][n-1];
    printf("%d\n",zbir);

    printf("Vnesete go brojot na redici, pa brojot na koloni na vtorata matrica: ");
    scanf("%d%d",&m,&n);
    printf("Vnesete ja matricata: \n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
```

```

scanf("%d",&b[i][j]);
int zbir1=0;
for(i=0;i<n;i++)
    zbir1+=b[0][i];
for(i=0;i<n;i++)
    zbir1+=b[m-1][i];
for(i=0;i<m;i++)
    zbir1+=b[i][0];
for(i=0;i<m;i++)
    zbir1+=b[i][n-1];
printf("%d\n",zbir1);

if(zbir>zbir1) {
    printf("Prvata matrica ima pogolem zbir.\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++)
            printf("%d ",a[i][j]);
        printf("\n");
    }
}
else {
    printf("Vtorata matrica ima pogolem zbir.\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++)
            printf("%d ",b[i][j]);
        printf("\n");
    }
}
return 0;
}

```

2. Матрица

Од тастатура се внесува матрица од цели броеви $A[m \times n]$, каде што димензиите m и n исто така се внесуваат од тастатура и не се поголеми од 100. Матрицата да се трансформира на следниов начин:

ако збирот на елементите на дадена редица од горната половина на матрицата е поголем од збирот на елементите на нејзината симетрична редица во однос на хоризонталата, тогаш редиците треба да си ги заменат местата.

Новодобиената матрица да се испечати на компјутерскиот екран. (ФИНКИ - Code – Пример испитни задачи 2014)

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main()
{
    int n,m,i,j,a[MAX][MAX];
    printf("Vnesete gi dimenziite na matricata: \n");
    scanf("%d%d",&m,&n);
    printf("Vnesete ja matricata: \n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);

    printf("\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++)
            printf("%d ",a[i][j]);
        printf("\n");
    }
    int sumi[MAX];
    int suma=0,pom;
    printf("\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++)
            suma+=a[i][j];
        sumi[i]=suma;
        suma=0;
    }
```

```

for(i=0;i<m/2;i++){
    for(j=0;j<n;j++){

        if(sumi[i]>sumi[m-i-1]){

            pom=a[i][j];
            a[i][j]=a[m-1-i][j];
            a[m-1-i][j]=pom;
        }

    }
}
printf("\n");
for(i=0;i<m;i++){
    for(j=0;j<n;j++)
        printf("%d ",a[i][j]);
    printf("\n");
}
return 0;
}

```

3. Да се напише програма во која се внесува квадратна матрица од цели броеви со максимална големина 10x10. Секој елемент од споредната дијагонала треба да се замени со збирот на елементите од споредната дијагонала под него. Средниот елемент не се менува, додека елементот во долниот лев агол ја добива вредноста на средниот елемент. На крај да се испечати новодобиената матрица. (ФЕИТ – 21.08.2013 , термин 1, група 2)

Влезна матрица:

8	3	2	1	<u>5</u>
2	3	3	<u>4</u>	4
5	6	<u>2</u>	4	8
9	<u>1</u>	11	43	12
<u>3</u>	14	15	2	16

Излезна матрица:

8	3	2	1	<u>10</u>
2	3	3	<u>6</u>	4
5	6	<u>2</u>	4	8
9	<u>3</u>	11	43	12
<u>2</u>	14	15	2	16

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10

int main()
{
    int i,j,n,a[MAX][MAX];
    printf("Vnesete ja dimenzijata na kvadratnata matrica: \n");
    scanf("%d",&n);

    printf("Vnesete ja matricata: \n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    for(i=0;i<n;i++){
        for(j=0;j<n;j++)
            printf("%d ",a[i][j]);
        printf("\n");
    }
    printf("\n");

    int sreden_element;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++)
        {
            if(i+j==n-1){
                if(i==j) sreden_element=a[i][j];
            }
        }
    }

    int suma=0,k,l;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            if(i+j==n-1){

                if((j==0 && i==(n-1))||(i==j))
                    a[i][j]=sreden_element;

                else {
```

```

        for(k=i+1;k<n;k++)
            for(l=0;l<n;l++) {
                if(k+l==n-1)
                    suma+=a[k][l];
            }
        a[i][j]=suma;
        suma=0;
    }
}
}
for(i=0;i<n;i++){
    for(j=0;j<n;j++)
        printf("%d ",a[i][j]);
    printf("\n");
}

return 0;
}

```

4. Во една метеоролошка станица во текот на 7 дена дневно се вршени по 5 мерењана температура на воздухот. Добиените податоци се внесуваат од тастатура и се сместуваат во дводимензионално поле. Да се најде и прикаже на екран максималната температура на воздух по ден.(ФЕИТ – 24.12.2012 термин 1 , група 1)

```

#include <stdio.h>
#include <stdlib.h>
#define MAX 10
int main()
{
    int i,j;
    int temperaturi[MAX][MAX];
    int max;
    printf("Vnesete gi temperaturite: \n");
    for(i=0;i<7;i++){// 7 dena
        for(j=0;j<5;j++){// 5 merenja dnevno
            scanf("%d",&temperaturi[i][j]);
        }
    }
    for(i=0;i<7;i++){
        max=temperaturi[i][0];
        for(j=0;j<5;j++)
        {
            if(temperaturi[i][j]>max) max=temperaturi[i][j];
        }
        printf("%d-tiot den ima maksimalna temperatura %d.\n",i+1,max);
    }
    return 0;
}

```

5. Да се напише програма која ќе пресмета средна вредност на елементите кои се наоѓаат истовремено под главна и под споредна дијагонала на една квадратна матрица.(ФЕИТ – 04.10.2008)

Пример:

8	3	2	1	5
2	3	3	4	4
5	6	2	4	8
9	1	11	43	12
3	14	15	2	16

средна вредност на броевите е: $(11+14+15+2)/4=10,5$

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main()
{
    int n,m,i,j,a[MAX][MAX];
    printf("Vnesete gi dimenziite na matricata: \n");
    scanf("%d%d",&m,&n);
    printf("Vnesete ja matricata: \n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);

    for(i=0;i<m;i++){
        for(j=0;j<n;j++)
            printf("%d ",a[i][j]);
        printf("\n");
    }
    printf("\n");
    int suma=0;int br=0;
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        {
            if((i>j) && (i+j>n-1)){
                suma+=a[i][j];
                br++;
            }
        }
    float sredna=suma/br;
    printf("Srednata vrednost na elementite koi se naogaat istovremeno pod glavnata i pod sporednata dijagonala e %.2f.",sredna);

    return 0;
}
```

6. Да се напише програма во која од тастатура се внесува квадратна матрица со димензии $n \times n$ (n се внесува од тастатура). Програмата, треба да ја пресмета сумата на сите елементи на матрицата. Потоа треба да ја измени матрицата во **долно триаголна**, на тој начин што сите елемент под главната дијагонала ќе добијат вредност еднаква на пресметаната сума. На крај треба да се испечати добиената матрица во формат на матрица. Максималната димензија на матрицата е 10×10 . (ФЕИТ – 05.09.2007)

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 10
int main()
{
    printf("Vnesete ja dimenzijata na kvadratnata matrica.\n");
    int n,i,j,a[MAX][MAX];
    int suma=0;
    scanf("%d",&n);
    printf("Vnesuvajte ja matricata: \n");
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            suma+=a[i][j];

    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            if(i>j) a[i][j]=suma;

    for(i=0;i<n;i++){
        for(j=0;j<n;j++)
            printf("%d ",a[i][j]);
        printf("\n");
    }
    return 0;
}
```

7. Да се напише функција која како влезни аргументи прима две матрици и низа од цели броеви. Функцијата треба преку низата (третиот аргумент во функцијата) да ги врати заедничките елементи во двете матрици (елементите кои се наоѓаат на иста позиција и се еднакви). (ФЕИТ 08.06.2011)

Пример:

Прва матрица:

1	9	1
5	2	7
9	6	11

Втора матрица:

14	9	1
8	2	72
91	6	1

Излез:

9	1	2	6
---	---	---	---

```
#include<stdio.h>
# define max 10
int n,br;
void func(int a[max][max],int b[max][max],int c[max])
{
    int i,j;
    for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
        if(a[i][j]==b[i][j])
        {
            c[br]=a[i][j];
            br++;
        }
    }
}
int main()
{
    int a[max][max],b[max][max],c[max];
    int i,j;
    printf("Vnesete ja dimenzijata za matricata:");
    scanf("%d",&n);
    printf("Vnesete gi elementite na pravata matrica:\n");
    for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
        printf("a[%d][%d]=",i,j);
        scanf("%d",&a[i][j]);
    }
    printf("Vnesete gi elementite na vtorata matrica:\n");
    for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
        printf("b[%d][%d]=",i,j);
        scanf("%d",&b[i][j]);
    }
    func(a,b,c);
    for(i=0;i<br;i++)
    printf("%d ",c[i]);
    return 0;
}
```

8. Да се напише програма во која од тастатура се внесуваат елементите (целобројни вредности) на матрица со димензија $m \times n$ (не поголеми од 100). Потребно е да се направи промена на матрицата така што максималниот елемент ќе ја добие вредноста на сумата на останатите елементи во таа редица. (ФЕИТ 10.06.2009)

Пример:

3	-2	5	1
-7	1	0	6
1	7	2	-8
4	-3	1	2

Излез:

3	-2	5	1
-7	1	0	6
1	-5	2	-8
4	-3	1	2

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main()
{
    int n,m,i,j,a[MAX][MAX];
    printf("Vnesete gi dimenziite na matricata: \n");
    scanf("%d%d",&m,&n);
    printf("Vnesete ja matricata: \n");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);

    printf("\n");
    for(i=0;i<m;i++){
        for(j=0;j<n;j++)
            printf("%d ",a[i][j]);
        printf("\n");
    }
    printf("\n");
    int maxi=0,maxj=0,max=a[0][0];

    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
        {
            if(a[i][j]>max){
                max=a[i][j];
                maxi=i;
                maxj=j;
            }
        }

    int suma=0;
    for(j=0;j<n;j++)
```

```
    suma+=a[maxi][j];  
a[maxi][maxj]=suma;  
  
for(i=0;i<m;i++){  
    for(j=0;j<n;j++){  
        printf("%d ",a[i][j]);  
        printf("\n");  
    }  
    return 0;  
}
```