

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <string.h>
#define MAX 30
/*5.1.a.Scrieți un program în C care citește numele și prenumele unei persoane și apoi
le afișează unul după altul pe același rând.
Utilizați initial funcția scanf()*/
/*
int main()
{
    char nume[10],prenume[10];
    printf("Numele:");
    scanf("%s",nume);
    printf("Prenume:");
    scanf("%s",prenume);
    printf("Nume+prenume:%s%s",nume,prenume);
    return 0;
}
*/

/*5.1.b.Scrieți un program în C care citește numele și prenumele unei persoane și apoi
le afișează unul după altul pe același rând.
Utilizați initial funcția scanf() pentru citire. Modificați apoi programul și
utilizați funcția gets() pentru citirea șirurilor de caractere. Pentru a vedea care
este diferența introduceți un nume sau prenume format din două cuvinte.*/
/*
int main()
{
    char nume[10],prenume[10];
    printf("Numele:");
    gets(nume);
    printf("Prenume:");
    gets(prenume);
    printf("Nume+prenume:%s%s",nume,prenume);
    return 0;
}
*/

/*5.2.a.Timpul se exprimă în ore, minute și secunde. Scrieți un program în C care
citește trei numere întregi care reprezintă orele, minutele și secunde și le
transformă în secunde */
/*
int main()
{
    int ore,minute,secunde;
    printf("Ore:");
    scanf("%d",&ore);
    printf("Minute:");

```

```

scanf("%d",&minute);
printf("Secunde:");
scanf("%d",&secunde);
secunde=secunde+minute*60+ore*3600;
printf("Totalul secundelor:%d\n",secunde);
return 0;
}
*/

```

/*5.2.b.Timpul se exprimă în ore, minute și secunde.

Cum arată programul dacă se introduce timpul în secunde și se cere transformarea lui în ore, minute și secunde?*/

```

/*
int main()
{
    int ore,minute,secunde;
    printf("Secunde:");
    scanf("%d",&secunde);
    ore=secunde/3600;
    minute=secunde%3600/60;
    secunde=secunde-minute*60-ore*3600;
    printf("Ore:%d,Minute:%d,Secunde:%d\n",ore,minute,secunde);
    return 0;
}
*/

```

/*5.3. La cronometrarea succesivă de patru ori a rulării unui program se obțin patru timpi diferiți, patru valori întregi. Să se calculeze și să se afișeze valoarea medie a timpului de execuție a programului.*/

```

/*
int main()
{
    float t1,t2,t3,t4,medie;
    printf("Timpul 1:");
    scanf("%f",&t1);
    printf("Timpul 2:");
    scanf("%f",&t2);
    printf("Timpul 3:");
    scanf("%f",&t3);
    printf("Timpul 4:");
    scanf("%f",&t4);
    medie=(t1+t2+t3+t4)/4;
    printf("Media este: %.2f",medie);
    return 0;
}
*/

```

/*5.4.Știindu-se valoarea curentă a tensiunii de alimentare a unui cuptor și abaterea admisă de la valoare nominală să se afișeze 1 în cazul în care este depășită abaterea permisă și 0 în caz contrar. Se va utiliza operatorul condițional.*/

```

/*

```

```

int main()
{
    float U,n;
    int a;
    printf("Tensiune:");
    scanf("%f",&U);
    printf("Valoarea nominala:");
    scanf("%f",&n);
    printf("Abaterea:");
    scanf("%d",&a);
    if (U>=(n-a)&&U<=(n+a))
        printf("\n1");
    else
        printf("\n0");
    return 0;
}
*/

/*5.5.Să se afișeze numărul de octeți (bytes) necesari reprezentării interne a fiecărui
tip de dată învățat (unsigned, int, long, float, double, char). Se va utiliza
operatorul sizeof.*/
/*
int main()
{
    printf("Dimensiuni:\n");
    printf("unsigned:%d\n",sizeof(unsigned));
    printf("int:%d\n",sizeof(int));
    printf("long:%d\n",sizeof(long));
    printf("float:%d\n",sizeof(float));
    printf("double:%d\n",sizeof(double));
    printf("char:%d\n",sizeof(char));
}
*/

/*5.6.Scrieți un program în C care convertește temperatura exprimată în grade
Fahrenheit (număr întreg), citită de la consolă, în grade Celsius, folosind
formula:*/
/*
int main()
{
    int fahrenheit;
    float celsius;
    printf("Temperatura in grade Fahrenheit:");
    scanf("%d",&fahrenheit);
    celsius=5*(fahrenheit-32)/9;
    printf("Temperatura in grade Celsius %.1f:",celsius);
    return 0;
}
*/

/*5.7.Știindu-se dimensiunile laturilor unui triunghi să se calculeze aria triunghiului,

```

Se va utiliza funcția `double sqrt(double)`, definită în biblioteca `math.h`. Se vor introduce doar valori care pot forma laturile unui triunghi.*/

```
/*
int main()
{
    int lat1,lat2,lat3;
    float p,arie;
    printf("Laturile triunghiului:\n");
    printf("Latura 1:");
    scanf("%d",&lat1);
    printf("Latura 2:");
    scanf("%d",&lat2);
    printf("Latura 3:");
    scanf("%d",&lat3);
    p=(lat1+lat2+lat3)/2;
    arie=sqrt(p*(p-lat1)*(p-lat2)*(p-lat3));
    printf("Aria este: %.1f",arie);
    return 0;
}
*/
```

/*6.5.În cadrul unui laborator se distribuie LED-uri studenților pentru a realiza experimentele. Știindu-se numărul total de LED-uri și numărul de studenți, câte LED-uri revin fiecărui student și câte LED-uri rămân nedistribuite? Se va ține cont de faptul că utilizatorul poate introduce orice fel de numere.*/

```
/*
int main()
{
    int l,s,i;
    i=0;
    while(i==0){
        printf("Introduceti numarul de LED-uri:");
        scanf("%d",&l);
        if (l<0)
            printf("Valoare eronata!");
        else
            i=1;
    }
    i=0;
    while(i==0){
        printf("Introduceti numarul de studenti:");
        scanf("%d",&s);
        if (l<=0)
            printf("Valoare eronata!");
        else
            i=1;
    }
    if (l>=s)
        printf("Numarul de led-uri per student %d ,Numarul de led-uri ramase este %d",l/s,l%s);
}
*/
```

```

else
    printf("Nu sunt destule led-uri");
return 0;
}
*/
/*6.6.La o licitație pentru cumpărarea unor plăci de dezvoltare Intel@ Galileo au depus oferte
trei firme. Să se determine care este cea mai bună ofertă.*/
/*
int main()
{
    float F1,F2,F3,r;
    printf("Introdu ofertele:");
    scanf("%f%f%f",&F1,&F2,&F3);
    if (F1>0&&F2>0&&F3>0)
        printf("Oferta cea mai buna:%.2f\n",((r=F1<F2?F1:F2),(r<F2?r:F2)));
    else
        printf("Valori eronate!\n");
    return 0;
}
*/
/*6.7.O firmă cumpără calculatoare și imprimante de la același furnizor în două etape. Știind
pentru fiecare etapă numărul de calculatoare, numărul de imprimante și suma plătită să se
specifice care este prețul unui calculator și prețul unei imprimante.*/
/*
int main()
{
    float p1,p2,pi,pc,det,detc,deti;
    int n1c,n2c,n1i,n2i;
    printf("Introduceti pret1 si pret2:");
    scanf("%f%f",&p1,&p2);
    printf("introduceti numarul de imprimante:");
    scanf("%d%d",&n1i,&n2i);
    printf("Introduceti numarul de calculatoare:");
    scanf("%d%d",&n1c,&n2c);
    if (n1c<=0||n2c<=0||n1i<=0||n2i<=0||p1<=0||p2<=0)
        printf("Valori eronate");
    else
    {
        det=n1i*n2c-n2i*n1c;
        if (det==0)
            printf("Sistem nedeterminat");
        else
        {
            pc=(n1i*p2-n2i*p1)/det;
            pi=(n2c*p1-n1c*p2)/det;
            printf("Pretul calculatoarelor:%f, Pretul imprimantelor:%f",pc,pi);
        }
    }
}

```

```

    }
    return 0;
}
*/
/*6.8.Presiunea dintr-o cameră este măsurată în trei momente diferite ale zilei. Să se
determine
variația maximă de presiune din acea zi.*/
/*
int main()
{
    int p1,p2,p3,max,min;
    printf("Presiunea 1:");
    scanf("%d",&p1);
    printf("Presiunea 2:");
    scanf("%d",&p2);
    printf("Presiunea 3:");
    scanf("%d",&p3);
    printf("Variatia maxima de presiune este
%d:",((max=p1>p2?p1:p2),(max>p3?max:p3))-((min=p1<p2?p1:p2),(min<p3?min:p3)));
}
*/
/*6.9.Să se determine dacă caracterul tastat este o vocală sau nu*/
/*
int main()
{
    char c;
    printf("Introduceti caracterul:");
    scanf("%c",&c);
    if (c=='A'||c=='E'||c=='I'||c=='O'||c=='U'||c=='a'||c=='e'||c=='i'||c=='o'||c=='u')
        printf("Caracterul introdus este o vocala.");
    else
        printf("Caracterul introdus nu este o vocala.");
    return 0;
}
*/
/*6.10.Știind Rata Specifică de Absorbție (SAR – Specific Absorption Rate) a unui telefon
mobil
să se precizeze dacă aceasta se încadrează în normele europene.
Obs. Se va verifica pe internet ce înseamnă SAR și care sunt normele europene în vigoare
(de exemplu: https://en.wikipedia.org/wiki/Specific\_absorption\_rate).*/
/*
int main()
{
    float SAR;
    printf("Introduceti Rata specifica de Absorbtie:");
    scanf("%f",&SAR);
    if (SAR>2)
        printf("Telefonul nu se incadreaza in normele europene");
}
*/

```

```

    else
        printf("Telefonul se incadreaza in normele europene");
    return 0;
}
*/
/*6.11.Pe baza specificării componentelor x și y ale direcției de deplasare, să se afișeze
direcția
de deplasare (punctul cardinal: N, NE, E, SE, S, SV, V, NV). In cazul în care x=0 și y=0
să se afișeze stationare.
Obs. Toate instrucțiunile if utilizate vor avea și partea de else.*/
/*
int main()
{
    int x,y;
    printf("x=");
    scanf("%d",&x);
    printf("y=");
    scanf("%d",&y);
    if (x==0&&y==0)
        printf("Stationare!");
    else if (x>0&&y==0)
        printf("E!");
    else if (x>0&&y>0)
        printf("NE!");
    else if (x==0&&y>0)
        printf("N!");
    else if (x<0&&y>0)
        printf("NV!");
    else if (x<0&&y==0)
        printf("V!");
    else if (x<0&&y<0)
        printf("SV!");
    else if (x==0&&y<0)
        printf("S!");
    else if (x>0&&y<0)
        printf("SE!");

    return 0;
}
*/
/*6.12.O mașină intră într-un oraș cu o anumită viteză și se îndreaptă spre un alt oraș aflat la
o
distanță cunoscută. Să se determine timpul necesar ca mașina să ajungă la destinație știind
că mașina poate să meargă uniform accelerat sau uniform decelerat.*/
/*
int main()
{
    int v,d;
    double t,delta,a;

```

```

printf("Introduceti Datele:\n");
printf("Viteza initiala:");
scanf("%d",&v);
printf("Distanța:");
scanf("%d",&d);
printf("Acceleratia:");
scanf("%lf",&a);
delta=v*v+4*(a/2)*d;
t=(-v+sqrt(delta))/a;
printf("Timpul este:%lf",t);
return 0;
}
*/

```

/*7.5.După închidere, într-un magazin de aparatură de medicală, se verifică încasările din ziua respectivă și se introduc rând pe rând în calculator sumele încasate pentru fiecare tranzacție introducându-se -1 cand s-au terminat valorile de introdus în calculator. Să se determine încasările din ziua respectivă.

```

int main()
{
    float i,t=0;
    do
    {
        printf("Introduceti suma incasata(-1 pentru a termina):");
        scanf("%f",&i);
        if (i!=-1)
            t=t+i;
    }
    while (i!=-1);
    printf("Suma totala:%.2f",t);
    return 0;
}
*/

```

/*7.6.Într-o secție de asamblare pe bandă a componentelor electronice fiecare muncitor are la dispoziție câte o cutie cu rezistențe. La un moment dat, știindu-se numărul de muncitori și numărul de rezistențe pe care îl mai are în cutie fiecare dintre ei, să se precizeze câte rezistențe se mai pot monta.*/

```

/*
int main()
{
    int i,t=0,n,r=1;
    printf("Nr. muncitori:");
    scanf("%d",&n);
    if (n<0)
        printf("Numar incorect");
    else
    {

```



```

        for (i=1; i<=n; i++)
        {
            do
            {
                printf("Nr.Rezistente pt muncitorul %d:",i);
                scanf("%d",&r);
            }
            while (r<0);
            t=t+r;
        }
        printf("Numarul total de rezistente disponibile:%d",t);
    }
    return 0;
}
*/

```

/*7.7.a. Să se afișeze un triunghi de 10 linii de steluțe utilizând instrucțiunea for:*/

```

/*
int main()
{
    int i,j,l=0;
    printf("Triunghiul afisat:\n");
    for (i=0;i<10;i++)
    {
        for (j=0;j<=i;j++)
            printf("*");
        printf("\n");
    }
}
*/

```

/*7.7.b. Să se afișeze un triunghi de 10 linii de steluțe utilizând instrucțiunea for: Modificați programul și citiți un număr întreg cuprins între 1 și 10 care reprezintă numărul de linii ale triunghiului de steluțe și afișați triunghiul. Ce se modifică dacă se dorește afișarea unui

triunghi cu vârful in jos?

```

/*
/*
int main()
{
    int i,j,l=0;
    while(l<1||l>10)
    {
        printf("Numarul de linii:");
        scanf("%d",&l);
    }
    printf("Triunghiul afisat:\n");
    for (i=l;i>=1;i--)

```

```

    {
        for (j=1;j<=i;j++)
            printf("**");
        printf("\n");
    }
}
*/
/*7.8.Dornic să afle consumul mediu al mașinii sale un șofer a înregistrat pe parcursul mai
multor
săptămâni cantitatea de combustibil cu care a alimentat și distanța parcursă cu combustibilul
respectiv. Să se scrie un program care calculează consumul mașinii pe fiecare perioadă
înregistrată precum și consumul mediu al mașinii pe toată durata înregistrărilor.
Modalitate de afișareși introducere a datelor:
Introduceți cantitatea de combustibil utilizata in litri (-1 pentru a termina): 25
Introduceți distanta parcursa (in km): 418
Consumul la aceasta alimentare a fost de 5.98 litri/100 km
Introduceți cantitatea de combustibil utilizata in litri (-1 pentru a termina): 16.2
Introduceți distanta parcursa (in km): 247
Consumul la aceasta alimentare a fost de 6.56 litri/100 km
Introduceți cantitatea de combustibil utilizata in litri (-1 pentru a termina): 9.5
Introduceți distanta parcursa (in km): 178
Consumul la aceasta alimentare a fost de 5.34 litri/100 km
Introduceți cantitatea de combustibil utilizata in litri (-1 pentru a termina): -1
Consumul mediu pe toata perioada a fost de 6.01 litri/100 km
*/
/*
int main()
{
    float c,ci,d,a=0,dt=0;
    int i=0;
    do
    {
        printf("Introduceți cantitatea de combustibil utilizata in litri(-1 pentru a termina):");
        scanf("%f",&c);
        if (c!=-1)
        {
            printf("Introduceți distansa parcursa (in km):");
            scanf("%f",&d);
            ci=c*100/d;
            printf("Consumul la aceasta alimentare a fost de %.2f litri/100km\n",ci);
            a=a+c;
            dt=dt+d;
            i++;
        }
    }
    while (c!=-1);
    if (i=0)
        printf("Nu sunt date;");
}

```

```

else
    printf("Consumul mediu pe toata perioada a fost de %.2f litri/100 km",a*100/dt);
}
*/
/*7.9.Să se scrie un program care calculează venitul brut săptămânal al fiecărui angajat
știind că
aceștia sunt plătiți cu ora în funcție de numărul de ore lucrate. Orele care depășesc 40 de
ore pe
săptămână sunt plătite cu o treime mai mult decât tariful orar.
Exemplu:
Introduceti numarul de ore lucrate (-1 pentru a termina): 25
Introduceti tariful orar al angajatului: 15
Venitul brut este: 375 lei
Introduceti numarul de ore lucrate (-1 pentru a termina): 40
Introduceti tariful orar al angajatului: 10
Venitul brut este: 400 lei
Introduceti numarul de ore lucrate (-1 pentru a termina): 43
Introduceti tariful orar al angajatului: 9
Venitul brut este: 396 lei
Introduceti numarul de ore lucrate (-1 pentru a termina): -1*/
/*
int main()
{
    float o,s,t;
    do
    {
        printf("Introduceti numarul de ore lucrate(-1 pentru a termina):");
        scanf("%f",&o);
        if (o!=-1)
        {
            printf("Introduceti tariful orar al angajatului:");
            scanf("%f",&t);
            if (o>40)
                s=40*t+(o-40)*((4*t)/3);
            else
                s=o*t;
            printf("Venitul brut este: %.2f lei\n",s);
        }
    }
    while (o!=-1);
}
*/
/*8.5.In urma măsurătorilor realizate în cadrul unor experimente se obțin mai multe seturi de
valori
pentru care este necesară calcularea mediei aritmetice. Să se scrie un program care
realizeaza
acest lucru. Pentru fiecare set de date se cunoaște numărul de valori din set și se consideră
că

```

datele introduse sunt corecte. Ce se schimbă dacă pentru fiecare set de date utilizatorul poate alege între calcularea mediei aritmetice sau a celei geometrice.

Obs. Se vor utiliza: o funcție pentru calculul mediei aritmetice și o funcție pentru calculul mediei

geometrice. Citirea valorilor experimentale se va face în funcții și afișarea rezultatelor se va face în funcția

```
main().*/
```

```
/*
```

```
float ma(int n);
```

```
float mg(int n);
```

```
int main()
```

```
{
```

```
    int n;
```

```
    char c;
```

```
    float r;
```

```
    do
```

```
    {
```

```
        printf("\nNumarul de valori:");
```

```
        scanf("%d",&n);
```

```
        if (n<=0)
```

```
        {
```

```
            printf("Numar de valori incorect.\n");
```

```
        }
```

```
        else
```

```
        {
```

```
            printf("A:media aritmetica; G:media geometrica; Raspuns:");
```

```
            c=getche();
```

```
            printf("\n");
```

```
            switch (c)
```

```
            {
```

```
            case 'A':
```

```
            {
```

```
                r=ma(n);
```

```
                printf("\nMedia aritmetica pentru cele %d valori este %f:",n,r);
```

```
                break;
```

```
            }
```

```
            case 'G':
```

```
            {
```

```
                r=mg(n);
```

```
                printf("\nMedia geometrica pentru cele %d valori este %f:",n,r);
```

```
                break;
```

```
            }
```

```
            default:
```

```
                printf("\nNu s-a ales o optiune valida.");
```

```
            }
```

```
        }
```

```

        printf("\nDoriti sa mai efectuati un calcul?Y=da;");
        c=getche();
    }
    while(c=='Y');
    return 0;
}
float mg(int n)
{
    int i;
    float m,p=1,a;
    for (i=0; i<n; i++)
    {
        printf("\nValuarea nr.%d este: ",i+1);
        scanf("%f",&a);
        p*=a;
    }
    m=pow(p,1/n);
    return m;
}
float ma(int n)
{
    int i;
    float m,s=0,a;
    for (i=0; i<n; i++)
    {
        printf("\nValuarea nr.%d este: ",i+1);
        scanf("%f",&a);
        s+=a;
    }
    m=s/n;
    return m;
}
*/

```

/*7.6.Să se scrie un program care permite, în funcție de dorința utilizatorului, să se calculeze valoarea rezistenței echivalente serie sau paralel a unui număr cunoscut de rezistențe, ale căror valori în ohmi () se introduc rând pe rând în calculator. Se consideră că datele introduse sunt corecte. Ce se modifică în program dacă se dorește calcularea rezistenței echivalente pentru mai multe seturi de rezistențe?

Obs. Se vor utiliza: o funcție pentru calculul rezistenței serie și o funcție pentru calculul rezistenței paralel.

Citirea valorilor rezistențelor se va face în funcții și afișarea rezultatelor se va face în funcția main().

*/

/*

```
float serie(int n)
```

```

{
    int i;
    float re=0,r;
    for (i=0; i<n; i++)
    {
        printf("\nRezistenta nr. %d este: ",i+1);
        scanf("%f",&r);
        re+=r;
    }
    return re;
}

float paralel(int n)
{
    int i;
    float re=0,r;
    for (i=0; i<n; i++)
    {
        printf("\nRezistenta nr. %d este: ",i+1);
        scanf("%f",&r);
        re+=1/r;
    }
    re=1/re;
    return re;
}

int main()
{
    int n;
    char c;
    float r;
    do
    {
        printf("\nNumar de rezistente:");
        scanf("%d",&n);
        if (n<=0)
        {
            printf("Numar de rezistente incorrect.\n");
        }
        else
        {
            printf("s:Serie;p:Paralel Raspuns:");
            c=getche();
            printf("\n");
            switch (c)
            {
                case 's':
                {
                    r=serie(n);
                    printf("\nRezistenta echivalenta pentru cele %d rezistente este %.2f:",n,r);
                }
            }
        }
    }
    while (c != 'p');
}

```

```

        break;
    }
    case 'p':
    {
        r=paralel(n);
        printf("\nRezistenta echivalenta pentru cele %d rezistente este %.2f:",n,r);
        break;
    }
    default:
        printf("\nNu s-a ales o optiune valida.");
    }
}
printf("\nDoriti sa mai efectuati un calcul?Y=da;");
c=getche();
}
while(c=='Y' || c=='y');
return 0;
}
*/

```

/*7.7.Să se determine în câte pătrate de arie maximă și de ce dimensiune se poate tăia o bucată de

tablă dreptunghiulară fără a pierde material. Se știe că dimensiunile bucății de tablă sunt valori

întregi exprimate în centimetri și dimensiunile pătratelor sunt de asemenea valori întregi în centimetri. Ce se modifică în program dacă se dorește împărțirea în pătrate a mai multor bucăți

de tablă? Se consideră că datele introduse sunt corecte.

Obs. Citirea dimensiunilor bucăților de tablă și afișarea rezultatelor se va face în funcția main().*/

/*

```
int cmmdc(int a,int b)
```

```

{
    int rest;
    rest = a % b;
    while (rest != 0)
    {
        a = b;
        b = rest;
        rest = a % b;
    }
    return b;
}

```

```

}
int main()
{

```

```

    char c;
    int a,b,nr;
    float r;
    do

```

```

{
    printf("\nLungime:");
    scanf("%d",&a);
    printf("Latime:");
    scanf("%d",&b);
    if (a<=0||b<=0)
    {
        printf("Valori eronate.\n");
    }
    else
    {
        printf("\n");
        nr=cmmdc(a,b);
        nr=(a/nr)*(b/nr);
        printf("Numarul de patrate de arie maxima este:%d",nr);
    }
    printf("\nDoriti sa mai efectuati un calcul?(Y=da)Raspuns:");
    c=getche();
}
while(c=='Y'||c=='y');
return 0;
}

```

*/

/*7.8.Specificându-se operația (+,-,*,/) și două numere reale să se afișeze expresia și rezultatul acesteia.

Obs. Se va utiliza câte o funcție pentru fiecare operație posibilă.Citirea operației și a numerelor și afișarea rezultatelor se va face în funcția main().

*/

/*

float adunare(float a, float b)

```

{
    return a+b;
}

```

float scadere(float a, float b)

```

{
    return a-b;
}

```

float inmultire(float a, float b)

```

{
    return a*b;
}

```

float impartire(float a, float b)

```

{
    return a/b;
}

```

int main()


```

{
    char c;
    float a,b;
    printf("Introduceti datele:");
    c=getche();
    scanf("%f%f",&a,&b);
    printf("\n");
    switch (c)
    {
        case '+':
        {
            printf("\nSuma celor 2 numere este:%.2f",adunare(a,b));
            break;
        }
        case '-':
        {
            printf("\nDiferenta celor 2 numere este:%.2f",scadere(a,b));
            break;
        }
        case '*':
        {
            printf("\nProdusul celor 2 numere este:%.2f",inmultire(a,b));
            break;
        }
        case '/':
        {
            printf("\nRezultatul impartirii celor 2 numere este:%.2f",impartire(a,b));
            break;
        }
        default:
            printf("\nNu s-a ales o optiune valida.");
    }
    return 0;
}

```

*/

/*9.3.Scrieți un program care citește de la tastatură două numere întregi și afișează factorialul

tuturor numerelor cuprinse între cele două valori citite, în ordine crescătoare.

Obs. Se va scrie o funcție care calculează factorialul unui număr întreg. Citirea celor două valorilor și afișarea rezultatelor se va face în funcția main().*/

/*

```

long long int fact(int n)
{
    int f=1,i;
    for (i=1; i<=n; i++)
        f*=i;
    return f;
}

```

```

int main()
{
    int n1,n2,i;
    printf("Primul numar:");
    scanf("%d",&n1);
    printf("Al doilea numar:");
    scanf("%d",&n2);
    if (n1<n2)
        for (i=n1; i<=n2; i++)
            printf("%d!=%lli\\t",i,fact(i));
    else
        for (i=n2; i<=n1; i++)
            printf("%d!=%lli\\t",i,fact(i));

    return 0;
}
*/

```

/*9.4.Scrieți un program care citește de la tastatură două numere întregi și afișează în ordine crescătoare toate numerele, cuprinse între cele două valori, care au suma cifrelor pară.

Obs. Se va scrie o funcție care calculează suma cifrelor unui număr întreg. Citirea celor două

valorilor și afișarea rezultatelor se va face în funcția main().*/

```

/*
int sum(int n)
{
    int s=0;
    while(n!=0)
    {
        s+=n%10;
        n=n/10;
    }
    return s;
}

int main ()
{
    int i,n1,n2,s;
    printf("Primul numar:");
    scanf("%d",&n1);
    printf("Al doilea numar:");
    scanf("%d",&n2);
    printf("\\nNumerele care au suma cifrelor para din intervalul[%d,%d]:\\n",n1,n2);
    if (n1<n2)
        for (i=n1; i<=n2; i++)
        {
            s=sum(i);
            if ((s%2)==0) printf(" %d",i);
        }
    else

```

```

        for (i=n2; i<=n1; i++)
        {
            s=sum(i);
            if ((s%2)==0) printf(" %d",i);
        }
    }
}
*/

```

/*9.5.Scrieți un program care citește de la tastatură două numere întregi și afișează în ordine descrescătoare toate numerele întregi, dintre cele două valori citite, a căror sumă a cifrelor este multiplu de 7.*/

/*

```

int sum(int n)
{
    int s=0;
    while(n!=0)
    {
        s+=n%10;
        n=n/10;
    }
    return s;
}

int main ()
{
    int i,n1,n2,s;
    printf("Primul numar:");
    scanf("%d",&n1);
    printf("Al doilea numar:");
    scanf("%d",&n2);
    printf("Numerele care au suma cifrelor multiplu de 7 din intervalul [%d,%d]:\n",n1,n2);
    if (n1<n2)
        for (i=n2; i>=n1; i--)
        {
            s=sum(i);
            if ((s%7)==0) printf(" %d",i);
        }
    else
        for (i=n1; i>=n2; i--)
        {
            s=sum(i);
            if ((s%7)==0) printf(" %d",i);
        }
}
*/

```

/*9.6.Scrieți un program care citește de la tastatură două numere întregi și afișează toate numerele prime dintre cele două valori citite, în ordine descrescătoare.

Obs. Se va scrie o funcție care determină dacă un număr întreg este prim sau nu. Citirea celor

două valorilor și afișarea rezultatelor se va face în funcția main ().*/

```

/*
int prim(int n)
{
    int c=0,i;
    for (i=1;i<=n/2;i++)
        if ((n%i)==0)
            c++;
    if (c==1)
        return 1;
    else
        return 0;
}
int main ()
{
    int i,n1,n2,c;
    printf("Primul numar:");
    scanf("%d",&n1);
    printf("Al doilea numar:");
    scanf("%d",&n2);
    printf("Numerele prime din intervalul [%d,%d]:\n",n1,n2);
    if (n1<n2)
        for (i=n2; i>=n1; i--)
        {
            if (prim(i)==1) printf(" %d",i);
        }
    else
        for (i=n1; i>=n2; i--)
        {
            if (prim(i)==1) printf(" %d",i);
        }
}
*/

```

/*9.7. Scrieți un program care citește de la tastatură două numere întregi și afișează în ordine crescătoare toate numerele dintre cele două valori care au suma divizorilor proprii pară.

Obs. Se va scrie o funcție care calculează suma divizorilor proprii ai unui număr întreg.

Citirea

celor două valorilor și afișarea rezultatelor se va face în funcția main(). */

```

/*
int nrdiv(int n)
{
    int s=0,i;
    for (i=1; i<n; i++)
        if ((n%i)==0)
            s+=i;
    return s-1;
}
int main ()
{

```

```

int i,n1,n2,s;
printf("Primul numar:");
scanf("%d",&n1);
printf("Al doilea numar:");
scanf("%d",&n2);
printf("Numerele care au suma divizorilor proprii para din intervalul [%d,%d]:\n",n1,n2);
if (n1<n2)
    for (i=n1; i<=n2; i++)
    {
        s=nrdiv(i);
        if ((s%2)==0) printf(" %d",i);
    }
else
    for (i=n2; i<=n1; i++)
    {
        s=nrdiv(i);
        if ((s%2)==0) printf(" %d",i);
    }
return 0;
}
*/

```

#define pi 3.14

/*10.1.Scrieți un program care citește de la tastatură două numere întregi și afișează cel mai mare divizor comun pentru fiecare pereche de numere întregi cuprinse între cele două valori citite.

Obs. Se va scrie o funcție care calculează cel mai mare divizor comun a două numere întregi. Citirea

celor două valorilor și afișarea rezultatelor se va face în funcția main().

Cel mai mare divizor comun in 3 variante de rezolvare(div1,div2,div3)

*/

/*

```

int div1(int a,int b);
int div2(int a,int b);
int div3(int a,int b);
int main()
{
    int a,b,i;
    char c;
    do
    {
        printf("Primul numar:");
        scanf("%d",&a);
        printf("Al doilea numar:");
        scanf("%d",&b);
        printf("Cel mai mare divizor comun dintre %d si %d este %d, %d, %d",a,b,div1(a,b),div2(a,b),div3(a,b));
        fflush(stdin);
        printf("\nDoriti sa repetati procesul D/N :");
    }
}

```

```

        scanf("%c",&c);
    }
    while(c=='D');
    return 0;
}
int div1(int a,int b)
{
    int i=0;
    while(a!=b)
    {
        if (a>b) a=a-b;
        else b=b-a;
        i++;
    }
    printf("(%d)",i);
    return a;
}
int div2(int a,int b)
{
    int t,i=0;
    while(b!=0)
    {
        t=a%b;
        a=b;
        b=t;
        i++;
    }
    printf("(%d)",i);
    return a;
}
int div3(int a,int b)
{
    if (b)
        div3(b,a%b);
    else
        return a;
}
*/

```

/*10.2.Să se scrie un program care permite calcularea ariilor unor suprafețe în funcție de forma lor

geometrică (pătrat, dreptunghi, triunghi, cerc).

Obs. Se va scrie o funcție pentru afișarea meniului, o funcție pentru calcularea și afișarea ariei unui

pătrat, o funcție pentru calcularea și afișarea ariei unui dreptunghi, o funcție pentru calcularea și

afișarea ariei unui triunghi, o funcție pentru calcularea și afișarea ariei unui cerc. Toate funcțiilor vor

fi apelate din funcția main().*/

```

/*
char meniu();
void patrat();
void drept();
void cerc();
void triunghi();
int main()
{
    char c;
    do
    {
        c=meniu();
        switch (c)
        {
            case 'P':
            {
                patrat();
                break;
            }
            case 'D':
            {
                drept();
                break;
            }
            case 'T':
            {
                triunghi();
                break;
            }
            case 'C':
            {
                cerc();
                break;
            }
            case 'I':
                break;
            default:
                printf("\nNu s-a ales o optiune valida.");
        }
    }while (c!='I');
    return 0;
}

char meniu()
{
    char c;
    printf("\nPrecizati forma geometrica a suprafetei apasand tasta corespunzatoare:\n");

```

```

    printf("P pentru patrat\nD pentru dreptunghi\nT pentru triunghi\nC pentru cerc\nI pentru
iesire\nRaspuns:");
    c=getche();
    return c;
}
void patrat()
{
    float i;
    printf("\nSpecificati latura suprafetei patrate in m:");
    scanf("%f",&i);
    printf("\nAria patratului este %.2f",i*i);
}
void drept()
{
    float l,L;
    printf("\nSpecificati laturile dreptunghiului:");
    scanf("%f %f",&l,&L);
    printf("\nAria dreptunghiului este %.2f",l*L);
}
void cerc()
{
    float r;
    printf("\nSpecificati raza suprafetei circulare in m:");
    scanf("%f",&r);
    printf("\nAria cercului este %.2f",pi*r*r);
}
void triunghi()
{
    double l1,l2,l3,sp,a;
    printf("\nLaturile triunghiului:");
    scanf("%lf %lf %lf",&l1,&l2,&l3);
    sp=(l1+l2+l3)/2;
    printf("\nAria triunghiului este %.2lf",sqrt(sp*(sp-l1)*(sp-l2)*(sp-l3)));
}
*/

```

/*10.3.Scrieți un program care citește de la tastatură două numere întregi și afișează în ordine

descrescătoare toate numerele, cuprinse între cele două valori, care au suma cifrelor divizibile cu 3 impară.

Obs. Se va scrie o funcție care calculează suma cifrelor divizibile cu 3 ale unui număr întreg.

Citirea

celor două valorilor și afișarea rezultatelor se va face în funcția main().*/

/*

```
int suma(int i);
```

```
int main()
```

```
{
```

```
    int a,b,i;
```



```

printf("Introduceti numerele:");
scanf("%d %d",&a,&b);
printf("\n");
if (a>b)
{
    for (i=a; i>=b;i--)
    {
        if (suma(i)%2!=0)
            printf("%d ",i);

    }
}
else
{
    for (i=b; i>=a;i--)
    {
        if (suma(i)%2!=0)
            printf("%d ",i);

    }
}
return 0;
}
int suma(int i)
{
    int s=0;
    while (i!=0)
    {
        if ((i%10)%3==0)
            s+=i%10;
        i=i/10;
    }
    return s;
}
*/

```

/*10.4.Scrieți un program care citește de la tastatură o succesiune de numere întregi nenule până la întâlnirea valorii de zero și afișează numărul care are cea mai mare valoare a sumei cifrelor.

Obs. Se va scrie o funcție care calculează suma cifrelor unui număr întreg. Citirea valorilor și afișarea

rezultatului se va face în funcția main().*/

/*

```

int suma(int i);
int main()
{
    int a=1,max=0,b,s;
    printf("Introduceti numerele:");
    while (a!=0)

```

```

{
    scanf("%d",&a);
    s=suma(a);
    if (max<s)
    {
        max=s;
        b=a;
    }
}
printf("Numarul este %d",b);
return 0;
}
int suma(int i)
{
    int s=0;
    while (i!=0)
    {
        s+=i%10;
        i=i/10;
    }
    return s;
}
*/

```

/* . Un semnal ECG analogic, care prezinta 6 cicluri cardiace, este discretizat și, ca urmare, valorile lui la intervale regulate de timp sunt introduse în calculator sub forma unui șir de valori. Să se determine toate valorile mai mari decât o valoare critică specificată, știind că valorile semnalului EKG pot fi pozitive, negative sau nule?

Obs. Se va scrie o funcție pentru introducerea valorilor în șir, o funcție pentru determinarea și afișarea

valorilor mai mari decât valoarea critică.*/

/*

void citire(int n,float a[])

```

{
    int i;
    printf("introduceti valorile:\n");
    for (i=0;i<n;i++)
    {
        scanf("%f",&a[i]);
    }
}

```

void afisare(int n,int c,float a[])

```

{
    int i;
    printf("\nValorile mai mari decat valoarea critica: ");
    for (i=0;i<n;i++)

```

```

    {
        if (a[i]>c||a[i]<(-c))
            printf("%.2f ",a[i]);
    }
}
int main()
{
    int n,c,a[MAX];
    printf("Valoarea critica:");
    scanf("%d",&c);
    printf("Introduceti numarul de valori:");
    scanf("%d",&n);
    if (n<1||n>MAX)
    {
        printf("Valori eronate!");
        return;
    }
    citire(n,a);
    afisare(n,c,a);
    return 0;
}
*/

```

/*11.2. Intr-un depozit de produse electrice există o zonă în care sunt depozitate cutiile cu becuri, fiecare cutie având un anumit număr de becuri în funcție de tipul acestora. Să se determine care este cutia cu cel mai mare număr de becuri, cea cu cel mai mic număr de becuri și numărul total de becuri existente în depozit. La final se cere așezarea cutiilor în ordinea descrescătoare a numărului de becuri? Obs. Se va scrie o funcție pentru introducerea numărului de becuri din cutii, o funcție pentru afișarea numărului de becuri din fiecare cutie, o funcție care determină cutia cu cel mai mic număr de becuri, o funcție care determină cutia cu cel mai mare număr de becuri, o funcție care determină numărul total de becuri, o funcție care sortează cutiile. Toate afișările se realizează în funcția main(), cu excepția afișării numărului de becuri din fiecare cutie care se realizează în funcția corespunzătoare.*/

```

/*
void citire(int n,int a[])
{
    int i;
    printf("introduceti numarul de becuri din fiecare cutie:\n");
    for (i=0; i<n; i++)
    {
        printf("Cutia %d:",i+1);
        scanf("%d",&a[i]);
    }
}

void afisare(int n,int a[])

```

```

{
    int i;
    printf("\nNumarul de becuri: ");
    for (i=0; i<n; i++)
    {
        printf("\nCutia %d: %d  ",i+1,a[i]);
    }
}
int max(int n,int a[])
{
    int i,m=a[0],c=0;
    for (i=1; i<n; i++)
    {
        if (m<a[i])
        {
            m=a[i];
            c=i;
        }
    }

    return c;
}
int min(int n,int a[])
{
    int i,m=a[0],c=0;
    for (i=0; i<n; i++)
    {
        if (m>a[i])
        {
            m=a[i];
            c=i;
        }
    }
    return c;
}
void ordonare(int n,int a[])
{
    int i,j,t;
    for (i=0; i<n-1; i++)
        for (j=i+1; j<n; j++)
            if (a[i]<a[j])
            {
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
}

```

```

int main()
{
    int n,c,a[50];
    printf("Introduceti numarul de cutii:");
    scanf("%d",&n);
    if (n<1||n>MAX)
    {
        printf("Valori eronate!");
        return;
    }
    citire(n,a);
    afisare(n,a);
    printf("\nCutia cu cele mai multe becuri %d",max(n,a)+1);
    printf("\nCutia cu numarul minim de becuri %d",min(n,a)+1);
    ordonare(n,a);
    afisare(n,a);
    return 0;
}

```

*/

/*11.3.Știind numărul de credite aferent fiecărei discipline, precum și notele obținute de un student la disciplinele respective să se calculeze media anuală ponderată a studentului respectiv.

Obs. Se va scrie o funcție pentru introducerea notelor obținute la discipline, o funcție pentru afișarea

notelor introduse, o funcție care calculează media ponderată, afișarea mediei se va face în funcția main().*/

/*

```

void citire(int n,float a[][MAX])
{
    int i;
    for (i=0;i<n;i++)
    {
        printf("Credite materia %d=",i+1);
        scanf("%f",&a[0][i]);
    }
    printf("Introduceti notele:\n");
    for (i=0;i<n;i++)
    {
        printf("Nota materia %d=",i+1);
        scanf("%f",&a[1][i]);
    }
}

void afisare(int n,float a[][MAX])
{
    int i;
    printf("\nNotele introduse sunt: ");
    for (i=0;i<n;i++)
    {

```

```

        printf("\nMateria %d credite:%.0f nota:%.2f ",i+1,a[0][i],a[1][i]);
    }
}
float medie(int n, float a[][MAX])
{
    int i;
    float s=0,m,t=0;
    for (i=0;i<n;i++)
    {
        s+=a[0][i]*a[1][i];
        t+=a[0][i];
    }
    m=s/t;
    return m;
}
int main()
{
    int n,c,a[2][MAX];
    printf("Numarul de materii:");
    scanf("%d",&n);
    if (n<1||n>MAX)
    {
        printf("Valori eronate!");
        return;
    }
    citire(n,a);
    afisare(n,a);
    printf("\nMedia ponderata este :%.2f",medie(n,a));
    return 0;
}
*/

```

/*11.4.Să se transforme un număr din baza 10 în baza 2, 8 și respectiv 16 folosind șiruri de caractere.

Obs. Se va scrie o funcție pentru punerea resturilor împărțirii la bază într-un șir de caractere și o funcție

care afișează în ordine inversă caracterele din șir.*/

/*

```

void baza2(char n[MAX],int a)
{
    int i=0,r;
    while(a)
    {
        r=a%2;
        n[i]=(r+'0');
        a/=2;
        i++;
    }
}

```

```

    n[i]='\0';
}
void baza8(char n[MAX],int a)
{
    int i=0,r;
    while(a)
    {
        r=a%8;
        n[i]=(r+'0');
        a/=8;
        i++;
    }
    n[i]='\0';
}
void baza16(char n[MAX],int a)
{
    int i=0,r;
    while(a)
    {
        r=a%16;
        a/=16;
        if (r<10) n[i]=(r+'0');
        else n[i]=(r-10+'A');
        i++;
    }
    n[i]='\0';
}
void baza(char n[MAX],int a,int b)
{
    int i=0,r;
    while(a)
    {
        r=a%b;
        a/=b;
        if (r<10) n[i]=(r+'0');
        else n[i]=(r+'A');
        i++;
    }
    n[i]='\0';
}
void afisare(char n[MAX])
{
    int m=strlen(n),i;
    printf("\n");
    for (i=m-1;i>-1;i--)
        printf("%c",n[i]);
}
int main()

```

```

{
    int a;
    char n2[MAX],n8[MAX],n16[MAX];
    printf("Numarul in baza 10:");
    scanf("%d",&a);
    baza2(n2,a);
    baza8(n8,a);
    baza16(n16,a);
    baza(n,a,2);
    printf("\nNumarul in baza 2:");
    afisare(n2);
    printf("\nNumarul in baza 2:");
    afisare(n);
    printf("\nNumarul in baza 8:");
    afisare(n8);
    printf("\nNumarul in baza 16:");
    afisare(n16);
    return 0;
}

```

*/

/*

12.1.Fiind dată o imagine în nuanțe de gri să se binarizeze această imagine folosind o valoare de prag specificată.

Obs. Se vor utiliza o funcție pentru citirea valorilor întregi ale matricii, o funcție pentru afișarea valorilor matricii și o funcție pentru binarizarea unei imagini. Toate afișările se vor face în funcția main().

*/

/*

```

void citire(int a,int b, int im[][MAX])
{
    int i,j;
    printf("Introduceti valorile matricii:\n");
    for (i=0; i<a; i++)
    {
        for (j=0; j<b; j++){
            printf("A[%d][%d]=",i,j);
            scanf("%d",&im[i][j]);
        }
    }
}

void scriere(int a,int b, int im[][MAX])
{

```



```

    int i,j;
    printf("\n Matricea este:\n");
    for (i=0; i<a; i++)
    {
        printf("\n");
        for (j=0; j<b; j++)
            printf("%5d",im[i][j]);
    }
}
void bin(int a,int b, int im[][MAX],int n)
{
    int i,j;
    for (i=0; i<a; i++)
        for (j=0; j<b; j++)
        {
            if (im[i][j]<n)
                im[i][j]=0;
            else
                im[i][j]=1;
        }
}
int main()
{
    int a,b,n,A[MAX][MAX];
    printf("Numarul de linii:");
    scanf("%d",&a);
    printf("Numarul de coloane:");
    scanf("%d",&b);
    if(a<0 || a>MAX || b<0 || b>MAX)
    {
        printf("Dimensiunile matricii sunt eronate");
        exit(1);
    }
    printf("Valuare de prag:");
    scanf("%d",&n);
    citire(a,b,A);
    bin(a,b,A,n);
    scriere(a,b,A);
    return 0;
}
*/
/*

```

Problema 12.2. Scrieți un program care citește dimensiunile și elementele reale ale unei matrici, o afișează și calculează și afișează media aritmetică a elementelor matricii. Ce se schimbă dacă se cere media aritmetică a elementelor de deasupra diagonalei principale.

Obs. Se vor utiliza: o funcție pentru citirea valorilor unei matrici, o funcție pentru afișarea valorilor unei matrici și o funcție pentru calcularea mediei aritmetice a elementelor unei matrici. Toate afișările se vor

face în funcția main().*/

/*

void citire(int a,int b, float A[][MAX])

{

int i,j;

printf("Introduceti valorile matricii:\n");

for (i=0; i<a; i++)

{

for (j=0; j<b; j++)

{

printf("A[%d][%d]=",i,j);

scanf("%f",&A[i][j]);

}

}

}

void scriere(int a,int b, float A[][MAX])

{

int i,j;

printf("\n Matricea este:\n");

for (i=0; i<a; i++)

{

printf("\n");

for (j=0; j<b; j++)

printf("%5.2f",A[i][j]);

}

}

float medie2(int a,int b, float A[][MAX])

{

int i,j,c=0;

float m,s=0.0;

for (i=0; i<a; i++)

for (j=i+1; j<b; j++)

{

c++;

s+=A[i][j];

}

m=s/c;

return m;

}

float medie(int a,int b, float A[][MAX])

{

int i,j;

```

float m,s=0;
for (i=0; i<a; i++)
    for (j=0; j<b; j++)
    {
        s+=A[i][j];
    }
m=s/(a*b);
return m;
}
int main()
{
    int a,b,n;
    float A[MAX][MAX];
    printf("Numarul de linii:");
    scanf("%d",&a);
    printf("Numarul de coloane:");
    scanf("%d",&b);
    if(a<0 || a>MAX || b<0 || b>MAX)
    {
        printf("dimensiunile matricii sunt eronate");
        exit(1);
    }
    citire(a,b,A);
    printf("Media tuturor elementelor este %.2f",medie(a,b,A));
    printf("\nMedia elementelor de deasupra coloanei principale este %.2f",medie2(a,b,A));
    scriere(a,b,A);
    return 0;
}

```

*/

/*

12.2 Doar cu media elementelor de deasupra diagonalei principale*/

/*

```

void matrice(int a, int b, float t[][30])

```

```

{

```

```

    int i, j;

```

```

    for(i=0; i<a; i++)

```

```

        for(j=0; j<b; j++)

```

```

        {

```

```

            printf("c[%d][%d]= ", i+1, j+1);

```

```

            scanf("%f", &t[i][j]);

```

```

        }

```

```

    }

```

```

void afisare(int a, int b, float t[][30])

```

```

{

```

```

    int i, j;

```

```

    for(i=0; i<a; i++)

```

```

    {

```

```

        for(j=0; j<b; j++)

```

```

        printf("%7.2f ", t[i][j]);
        printf("\n");
    }
}
float medie(int a, int b, float m[][30]);
int main()
{
    int nl, nc;
    float t[MAX][MAX];
    printf("Introduceti numarul de linii:");
    scanf("%d", &nl);
    printf("Introduceti numarul de coloane:");
    scanf("%d", &nc);
    if(nl<0 || nl>MAX || nc<0 || nc>MAX)
    {
        printf("dimensiunile matricii sunt eronate");
        exit(1);
    }
    matrice(nl, nc, t);
    afisare(nl, nc, t);
    printf("Media aritmetica a elementelor de deasupra diagonalei principale:%.2f", medie(nl,
nc, t));
    return 0;
}
float medie(int a, int b, float m[][MAX])
{
    int i, j, contor=0;
    float s=0;
    for(i=0; i<a; i++)
        for(j=0; j<b; j++)
        {
            if (j>i)
            {
                s=s+m[i][j];
                contor++;
            }
        }
    s=s/contor;
    return s;
}
*/
/*

```

Problema 12.3. Scrieți un program care citește dimensiunile și elementele întregi ale unei matrici a, o

afișează și calculează și afișează matricea b ale cărei elemente se calculează astfel:

$b_{ij} = \text{suma cifrelor numărului } a_{ij}$

Obs. Se vor utiliza: o funcție pentru citirea valorilor unei matrici, o funcție pentru afișarea valorilor unei

matrici, o funcție care calculează suma cifrelor unui număr întreg și o funcție pentru calcularea noii matrici. Toate afișările se vor face în funcția main().*/

/*

```
void citire(int a,int b, int A[][MAX])
```

```
{
    int i,j;
    printf("Introduceti valorile matricii:\n");
    for (i=0; i<a; i++)
    {
        for (j=0; j<b; j++)
        {
            printf("A[%d][%d]=",i,j);
            scanf("%d",&A[i][j]);

        }
    }
}
```

```
void scriere(int a,int b, int A[][MAX])
```

```
{
    int i,j;
    for (i=0; i<a; i++)
    {
        printf("\n");
        for (j=0; j<b; j++)
            printf("%5d",A[i][j]);
    }
}
```

```
int suma(int a)
```

```
{
    int s=0;
    do
    {
        s+=a%10;
        a=a/10;
    }
    while(a!=0);
    return s;
}
```

```
void creare(int a,int b, int A[][MAX],int B[][MAX])
```

```
{
    int i,j;
    for (i=0; i<a; i++)
        for (j=0; j<b; j++)
        {
            B[i][j]=suma(A[i][j]);
        }
}
```

```

    }
}
int main()
{
    int a,b,n,A[MAX][MAX],B[MAX][MAX];
    printf("Numarul de linii:");
    scanf("%d",&a);
    printf("Numarul de coloane:");
    scanf("%d",&b);
    if(a<0 || a>MAX || b<0 || b>MAX)
    {
        printf("dimensiunile matricii sunt eronate");
        exit(1);
    }
    citire(a,b,A);
    printf("\n Matricea A este:\n");
    scriere(a,b,A);
    creare(a,b,A,B);
    printf("\n Matricea B este:\n");
    scriere(a,b,B);
    return 0;
}
*/
/*

```

12.4. Scrieți un program care citește dimensiunea și elementele reale ale unei matrici pătratice, o afișează și calculează și afișează produsul elementelor de pe diagonala secundară.

Obs. Se vor utiliza: o funcție pentru citirea valorilor unei matrici, o funcție pentru afișarea valorilor unei matrici și o funcție pentru calcularea produsului elementelor de pe diagonala secundară a unei matrici.

Toate afișările se vor face în funcția main().

```

*/
/*
void citire(int a, float A[][MAX])
{
    int i,j;
    printf("Introduceti valorile matricii:\n");
    for (i=0; i<a; i++)
    {
        for (j=0; j<a; j++)
        {
            printf("A[%d][%d]=",i,j);
            scanf("%f",&A[i][j]);
        }
    }
}
}

```

```

}
void scriere(int a, float A[][MAX])
{
    int i,j;
    printf("\n Matricea este:\n");
    for (i=0; i<a; i++)
    {
        printf("\n");
        for (j=0; j<a; j++)
            printf("%5.2f",A[i][j]);
    }
}
float produs(int a, float A[][MAX])
{
    int i,j;
    float s=1;;
    for (i=0; i<a; i++)
        for (j=0; j<a; j++)
        {
            if ((i+j)==(a-1))
            {
                s*=A[i][j];
            }
        }
    return s;
}
int main()
{
    int a;
    float A[MAX][MAX];
    printf("Numarul de linii si coloane:");
    scanf("%d",&a);
    if(a<0|| a>MAX)
    {
        printf("Dimensiunile matricii sunt eronate");
        exit(1);
    }
    citire(a,A);
    printf("Produsul elementelor de pe diagonala secundara este %.2f",produs(a,A));
    scriere(a,A);
    return 0;
}
*/
/*

```

12.5.Scrieți un program care citește dimensiunile și elementele reale ale unei matrici, o afișează și calculează și afișează transpusa matricii.

Obs. Se vor utiliza: o funcție pentru citirea valorilor unei matrici, o funcție pentru afișarea valorilor unei

matrici și o funcție pentru calcularea transpusei unei matrici. Toate afișările se vor face în funcția main()*/

```
/*
void citire(int a,int b, float A[][MAX])
{
    int i,j;
    printf("Introduceti valorile matricii:\n");
    for (i=0; i<a; i++)
    {
        for (j=0; j<b; j++)
        {
            printf("A[%d][%d]=",i,j);
            scanf("%f",&A[i][j]);
        }
    }
}

void scriere(int a,int b, float A[][MAX])
{
    int i,j;
    for (i=0; i<a; i++)
    {
        printf("\n");
        for (j=0; j<b; j++)
            printf("%5.2f",A[i][j]);
    }
}

void creare(int a,int b, float A[][MAX],float B[][MAX])
{
    int i,j;
    for (i=0; i<a; i++)
        for (j=0; j<b; j++)
        {
            B[j][i]=A[i][j];
        }
}

int main()
{
    int a,b,n;
    float A[MAX][MAX],B[MAX][MAX];
    printf("Numarul de linii:");
    scanf("%d",&a);
    printf("Numarul de coloane:");
    scanf("%d",&b);
    if(a<0 || a>MAX || b<0 || b>MAX)
    {
        printf("Dimensiunile matricii sunt eronate");
    }
}
```



```

        exit(1);
    }
    citire(a,b,A);
    creare(a,b,A,B);
    printf("\n Matricea A este:\n");
    scriere(a,b,A);
    printf("\n Matricea A transpusa este:\n");
    scriere(a,b,B);
    return 0;
}
*/

```

/*13.1.Scrieți un program care citește numărul elementelor și elementele întregi a două șiruri de numere a și b, le afișează și calculează și afișează, ordonat descrescător, șirul de numere c ale cărei elemente se calculează astfel:

ci= suma cifrelor numerelor ai si bi

Obs. Se vor utiliza: o funcție pentru citirea unui șir de numere întregi, o funcție pentru afișarea unui șir

de numere întregi, o funcție pentru calcularea sumei cifrelor unui număr întreg, o funcție pentru calculul

noului șir și o funcție pentru ordonarea descrescătoare a unui șir. Toate afișările se fac în funcția

main().

*/

/*

```

void citire(int n,int a[MAX])

```

```

{
    int i;
    printf("Introduceti elementele sirului:\n");
    for (i=0;i<n;i++)
    {
        printf("  sir[%d]=",i);
        scanf("%d",&a[i]);
    }
}

```

```

void afisare(int n,int a[MAX])

```

```

{
    int i;
    for (i=0;i<n;i++)
    {
        printf("  sir[%d]=%d",i,a[i]);
    }
}

```

```

int suma(int e)

```

```

{
    int i,s=0;
    while(e!=0)
    {

```

```

        s+=e%10;
        e=e/10;
    }
    return s;
}
void calcul(int n,int a[MAX],int b[MAX],int c[MAX])
{
    int i,j;
    for (i=0;i<n;i++)
    {
        c[i]=suma(a[i])+suma(b[i]);
    }
}
void ordonare(int n,int a[MAX])
{
    int i,j,k;
    for (i=0;i<n;i++)
    {
        for (j=i;j<n;j++)
            if (a[i]<a[j])
            {
                k=a[i];
                a[i]=a[j];
                a[j]=k;
            }
    }
}

```

```

int main()
{
    int n,m;
    int a[MAX],b[MAX],c[MAX];
    printf("Nr. de elemente ale sirului a:");
    scanf("%d",&n);
    printf("Nr. de elemente ale sirului b:");
    scanf("%d",&m);
    if (n!=m||n<1||n>MAX||m<1||m>MAX)
    {
        printf("numar inegal de elemente!");
        return 0;
    }
    citire(n,a);
    citire(n,b);
    calcul(n,a,b,c);
    ordonare(n,c);
    printf("Elementele sirului c in ordine descrescatoare:\n");
    afisare(n,c);
}

```

```

    return 0;
}
*/
/*13.2.Scrieți un program care citește numărul elementelor și elementele întregi ale unui șir
de
numere a, îl afișează și calculează și afișează numărul cifrelor de 5 aflate după primul
număr par din șir.
Obs. Se vor utiliza: o funcție pentru citirea unui șir de numere întregi, o funcție pentru
afișarea unui șir
de numere întregi, o funcție pentru calculul poziției primului număr par din șir, o funcție
pentru calcularea
numărului cifrelor de 5 ale unui număr întreg și o funcție pentru calcularea numărului cifrelor
de 5 după
primul număr par din șir. Toate afișările se fac în funcția main(). Se va lua în considerare și
cazul în care
nu există un număr par în șir.*/
/*
void citire(int n,int a[MAX])
{
    int i;
    printf("Introduceti elementele sirului:\n");
    for (i=0;i<n;i++)
    {
        printf("  sir[%d]=",i);
        scanf("%d",&a[i]);
    }
}
void afisare(int n,int a[MAX])
{
    int i;
    printf("Elementele sirului sunt: ");
    for (i=0;i<n;i++)
    {
        printf("  a[%d]=%d",i,a[i]);
    }
}
int par(int n,int a[MAX])
{
    int i;
    for (i=0;i<n;i++)
    {
        if (a[i]%2==0) return i;
    }
    return 0;
}
int nr(int e)
{
    int i,c=0;

```

```

while(e!=0)
{
    if (e%10==5) c++;
    e=e/10;
}
return c;
}
int nrsir(int n,int k,int a[MAX])
{
    int i,c=0;
    for (i=k+1;i<n;i++)
    {
        c+=nr(a[i]);
    }
    return c;
}
int main()
{
    int n,m;
    int a[MAX];
    printf("Nr. de elemente ale sirului a:");
    scanf("%d",&n);
    if (n<1||n>MAX)
    {
        printf("Valoare eronata;");
        exit(1);
    }
    citire(n,a);
    afisare(n,a);
    m=par(n,a);
    if (m||(a[0]%2==0))
    {
        printf("\nNumarul de cifre 5 dupa primul numar par este %d;\n",nrsir(n,m,a));
    }
    else
        printf("\nSirul nu are nici un numar par in el;\n");
    return 0;
}
*/

```

/*13.3.Scrieți un program care citește numărul elementelor și elementele reale ale unui șir de numere a, afișează șirul citit și calculează, ordonează crescător și afișează șirul de numere b ale cărui elemente se calculează astfel:

$b_i = 2a_i + p$, $a_i < 0$

$b_i = a_i$, $a_i = 0$

$b_i = a_i^2$, $a_i > 0$

Obs. Se vor utiliza: o funcție pentru citirea unui șir de numere reale, o funcție pentru afișarea unui

șir de numere reale, o funcție pentru calculul elementelor noului șir și o funcție pentru ordonarea

crescătoare a unui șir. Toate afișările se fac în funcția main().*/

/*

void citire(int n,float a[MAX])

```
{
    int i;
    printf("Introduceti elementele sirului:\n");
    for (i=0;i<n;i++)
    {
        printf("  sir[%d]=",i);
        scanf("%f",&a[i]);
    }
}
```

void afisare(int n,float a[MAX],char C)

```
{
    int i;
    for (i=0;i<n;i++)
    {
        printf(" %c[%d]=%.2f",C,i,a[i]);
    }
}
```

void calcul(int n,float a[MAX],float b[MAX])

```
{
    int i;
    for (i=0;i<n;i++)
    {
        if (a[i]==0)
        {
            b[i]=a[i];
        }
        else
        {
            if (a[i]<0) b[i]=2*a[i]+9;
            else b[i]=a[i]*a[i];
        }
    }
}
```

void ordonare(int n,float a[MAX])

```
{
    int i,j,k;
    for (i=0;i<n;i++)
    {
        for (j=i;j<n;j++)
            if (a[i]>a[j])
            {
                k=a[i];
                a[i]=a[j];
                a[j]=k;
            }
    }
}
```

```

        a[j]=k;
    }
}
}

int main()
{
    int n,m;
    float a[MAX],b[MAX];
    printf("Nr. de elemente ale sirului a:");
    scanf("%d",&n);
    if (n<1||n>MAX)
    {
        printf("Valoare eronata;");
        exit(1);
    }
    citire(n,a);
    printf("\nElementele sirului citit sunt:");
    afisare(n,a,'a');
    calcul(n,a,b);
    printf("\nElementele sirului calculat sunt:");
    afisare(n,b,'b');
    ordonare(n,b);
    printf("\n(Dupa ordonare)");
    afisare(n,b,'b');
    return 0;
}
*/

```

/*13.4.Scrieți un program care citește dimensiunile și elementele întregi ale unei matrici a, o afișează și calculează și afișează matricea b ale cărei elemente se calculează astfel:

$b_{ij} = \text{suma cifrelor impare ale lui } a_{ij}$

Obs. Se vor utiliza: o funcție pentru citirea valorilor unei matrici, o funcție pentru afișarea valorilor unei

matrici, o funcție pentru calcularea sumei cifrelor impare ale unui număr întreg și o funcție pentru calculul

noii matrici. Toate afișările se fac în funcția main().*/

/*

```

void citire(int n,int m, int A[][MAX])
{
    int i,j;
    printf("Introduceti valorile matricii:\n");
    for (i=0; i<n; i++)
    {

        for (j=0; j<m; j++){
            printf("A[%d][%d]=",i,j);
            scanf("%d",&A[i][j]);

```

```

    }
    }
}
void scriere(int n,int m, int A[][MAX])
{
    int i,j;
    printf("\n Matricea este:\n");
    for (i=0; i<n; i++)
    {
        printf("\n");
        for (j=0; j<m; j++)
            printf("%5d",A[i][j]);
    }
}
int suma(int e)
{
    int i,s=0;
    while(e!=0)
    {
        if ((e%10)%2!=0)
            s+=e%10;
        e=e/10;
    }
    return s;
}
void calcul(int n,int m, int a[][MAX],int b[][MAX])
{
    int i,j;
    for (i=0; i<n; i++)
        for (j=0; j<m; j++)
            b[i][j]=suma(a[i][j]);
}

int main()
{
    int m,n,a[MAX][MAX],b[MAX][MAX];
    printf("Numarul de linii:");
    scanf("%d",&n);
    printf("Numarul de coloane:");
    scanf("%d",&m);
    if(n<0 || n>MAX || m<0 || m>MAX)
    {
        printf("Dimensiunile matricii sunt eronate");
        exit(1);
    }

    citire(n,m,a);
    calcul(n,m,a,b);
}

```

```

        scriere(n,m,b);
    return 0;
}
*/
/*13.5.Scrieți un program care citește dimensiunea și elementele reale ale unei matrici
pătratică, afișează matricea și calculează și afișează media aritmetică a elementelor de
deasupra diagonalei principale.
Obs. Se vor utiliza: o funcție pentru citirea unei matrici pătratică, o funcție pentru afișarea
unei matrici
pătratică, o funcție pentru calcularea mediei aritmetice a elementelor de deasupra diagonalei
principale a
unei matrici. Toate afișările se fac în funcția main().*/
/*
void citire(int n,float A[][MAX])
{
    int i,j;
    printf("Introduceti valorile matricii:\n");
    for (i=0; i<n; i++)
    {
        for (j=0; j<n; j++)
        {
            printf("A[%d][%d]=",i,j);
            scanf("%f",&A[i][j]);
        }
    }
}
void scriere(int n,float A[][MAX])
{
    int i,j;
    printf("\n Matricea este:\n");
    for (i=0; i<n; i++)
    {
        printf("\n");
        for (j=0; j<n; j++)
            printf("%5.1f",A[i][j]);
    }
}
float medie(int n,float A[][MAX])
{
    int i,j;
    float m,s=0;
    for (i=0; i<n; i++)
        for (j=i+1; j<n; j++)
            s+=A[i][j];
    m=s/(n*(n-1)/2);
    return m;
}
int main()

```



```

{
    int n;
    float a[MAX][MAX];
    printf("Numarul de linii si coloane:");
    scanf("%d",&n);
    if(n<0|| n>MAX )
    {
        printf("Dimensiunile matricii sunt eronate");
        exit(1);
    }
    citire(n,a);
    scriere(n,a);
    printf("\nMedia aritmetica a numerelor de deasupra diagonalei principale
este: %.2f",medie(n,a));
    return 0;
}
*/
/*14.1.Scrieti un program care citește dimensiunea și elementele reale ale unei matrici
pătratică
a, o afișează și calculează și afișează matricea b ale cărei elemente se calculează astfel:
 $b_{ij}=2*a_{ij}-7$ , dacă  $a_{ij}$  este sub diagonala principală;
 $b_{ij}=a_{ij}^2+2$ , în rest
Obs. Se vor utiliza: o funcție pentru citirea valorilor unei matrici, o funcție pentru afișarea
valorilor unei
matrici și o funcție pentru calcularea noii matrici. Toate afișările se fac în funcția main().*/
/*
void citire(int n,float A[][MAX])
{
    int i,j;
    printf("\nIntroduceti valorile matricii:\n");
    for (i=0; i<n; i++)
    {

        for (j=0; j<n; j++){
            printf("M[%d][%d]=",i,j);
            scanf("%f",&A[i][j]);

        }
    }
}

void scriere(int n, float A[][MAX])
{
    int i,j;
    for (i=0; i<n; i++)
    {
        printf("\n");
        for (j=0; j<n; j++)
            printf("%5.1f",A[i][j]);
    }
}

```

```

    }
}
void calcul(int n,float A[][MAX],float B[][MAX])
{
    int i,j;
    for (i=0; i<n; i++)
    {
        for (j=0; j<n; j++)
            if (j<i)
                B[i][j]=2*A[i][j]-7;
            else
                B[i][j]=A[i][j]*A[i][j]+2;
    }
}
int main()
{
    int n;
    float a[MAX][MAX],b[MAX][MAX];
    printf("Numarul de linii si coloane:");
    scanf("%d",&n);
    if (n<1||n>MAX)
    {
        printf("Valori eronate.");
        exit(1);
    }
    citire(n,a);
    printf("\nMatricea A:\n");
    scriere(n,a);
    calcul(n,a,b);
    printf("\nMatricea B:\n");
    scriere(n,b);
    return 0;
}
*/

```

/*14.2.Scrieți un program care citește dimensiunile și elementele întregi a două matrici a și b, le afișează și calculează și afișează matricea c ale cărei elemente se calculează astfel:

$c_{ij} = c_{mmdc}(a_{ij}, b_{ij})$;

Obs. Se vor utiliza: o funcție pentru citirea valorilor unei matrici, o funcție pentru afișarea valorilor unei

matrici, o funcție care calculează cel mai mare divizor comun a două numere întregi și o funcție pentru

calcularea noii matrici. Toate afișările se vor face în funcția main().*/

/*

```

void citire(int n,int m, int A[][MAX])
{
    int i,j;
    printf("\nIntroduceti valorile matricii:\n");
    for (i=0; i<n; i++)

```

```

{
    for (j=0; j<m; j++){
        printf("M[%d][%d]=",i,j);
        scanf("%d",&A[i][j]);
    }
}
}
void scriere(int n,int m, int A[][MAX])
{
    int i,j;
    for (i=0; i<n; i++)
    {
        printf("\n");
        for (j=0; j<m; j++)
            printf("%5d",A[i][j]);
    }
}
int cmmdc(int a,int b)
{
    while(a!=b)
    {
        if (a>b) a=a-b;
        else b=b-a;
    }
    return a;
}
void calcul(int n,int m,int A[][MAX],int B[][MAX],int C[][MAX])
{
    int i,j;
    for (i=0; i<n; i++)
    {
        for (j=0; j<m; j++)
            C[i][j]=cmmdc(A[i][j],B[i][j]);
    }
}
int main()
{
    int m,n,a[MAX][MAX],b[MAX][MAX],c[MAX][MAX];
    printf("Numarul de linii:");
    scanf("%d",&n);
    printf("Numarul de coloane:");
    scanf("%d",&m);
    if (n<1||n>MAX||m<1||m>MAX)
    {
        printf("Valori eronate.");
        exit(1);
    }
}

```

```

    }
    citire(n,m,a);
    printf("\nMatricea A:\n");
    scriere(n,m,a);
    citire(n,m,b);
    printf("\nMatricea B:\n");
    scriere(n,m,b);
    calcul(n,m,a,b,c);
    printf("\nMatricea C:\n");
    scriere(n,m,c);
    return 0;
}

```

```

*/

```

/*14.3.Scrieți un program care citește numărul elementelor și elementele întregi ale unui șir de

numere a, îl afișează și calculează și afișează numărul cifrelor de 3 aflate după primul număr prim din șir.

Obs. Se vor utiliza: o funcție pentru citirea unui șir de numere întregi, o funcție pentru afișarea unui șir

de numere întregi, o funcție pentru calculul poziției primului număr prim din șir, o funcție pentru

calcularea numărului cifrelor de 3 ale unui număr întreg și o funcție pentru calcularea numărului cifrelor

de 3 aflate după primul număr prim din șir. Toate afișările se fac în funcția main(). Se va lua în considerare

și cazul în care nu există un număr prim în șir.*/

```

/*

```

```

void citire(int n,int a[MAX])
{
    int i;
    printf("Introduceti elementele sirului:\n");
    for (i=0;i<n;i++)
    {
        printf("  sir[%d]=",i);
        scanf("%d",&a[i]);
    }
}

```

```

void afisare(int n,int a[MAX])
{
    int i;
    printf("Elementele sirului sunt: ");
    for (i=0;i<n;i++)
    {
        printf("  a[%d]=%d",i,a[i]);
    }
}

```

```

int prim(int n,int a[MAX])

```

```

{
    int c,i,j;
    for (i=0;i<n;i++)
    {
        c=0;
        if (a[i]==1)
            return i;
        else
        {
            for (j=1;j<=a[i]/2;j++)
                if ((a[i]%j)==0)
                    c++;
            if (c==1)
                return i;
        }
    }
    return n;
}

int nr(int e)
{
    int i,c=0;
    while(e!=0)
    {
        if (e%10==3) c++;
        e=e/10;
    }
    return c;
}

int nrsir(int n,int k,int a[MAX])
{
    int i,c=0;
    for (i=k+1;i<n;i++)
    {
        c+=nr(a[i]);
    }
    return c;
}

int main()
{
    int n,m;
    int a[MAX];
    printf("Nr. de elemente ale sirului a:");
    scanf("%d",&n);
    if (n<1||n>MAX)
    {
        printf("Valoare eronata;");
        exit(1);
    }
}

```

```
}  
citire(n,a);  
afisare(n,a);  
m=prim(n,a);  
printf("M este:%d",m);  
printf("\nNumarul de cifre 3 dupa primul numar prim este %d;\n",nrsir(n,m,a));  
return 0;  
}  
*/
```