

5. Алгоритми и програми за работа с двумерни масиви

Задача е_1: Да се състави програма със следното условие: Даден е двумерен масив $C(M,N)$ с произволни въведени елементи от целочислен тип. Да се намери средноаритметичната стойност на елементите, които са по-големи от 0. Да се изведе полученият резултат.

Решение: Стойностите за брой редове и стълбове могат да се въведат от клавиатурата. За конкретната задача се използват константни стойности за $M=3$ и $N=4$, зададени в програмата, което е по-кратко, но не е често приложимо.

За работа с двумерния масив се реализират две вложени циклични обработки с броячи i и j , изменящи стойностите на индексите за ред и стълб. Действията са въвеждане на стойност за текущия елемент и проверка по зададеното условие, за да се намира сумата и броя.

За пресмятане на средната стойност, за да се избегне делене на 0, се проверява броят дали има намерени елементи. Изчислява се средноаритметичната стойност и се извежда резултатът или извежда съобщение за грешка.

Използват се следните променливи:

c – име на двумерен масив;

M, N – константи за броя на редовете и стълбовете;

i, j – управляващи променливи за цикли и текущи индекси
съответно за ред и стълб на елемент от масива;

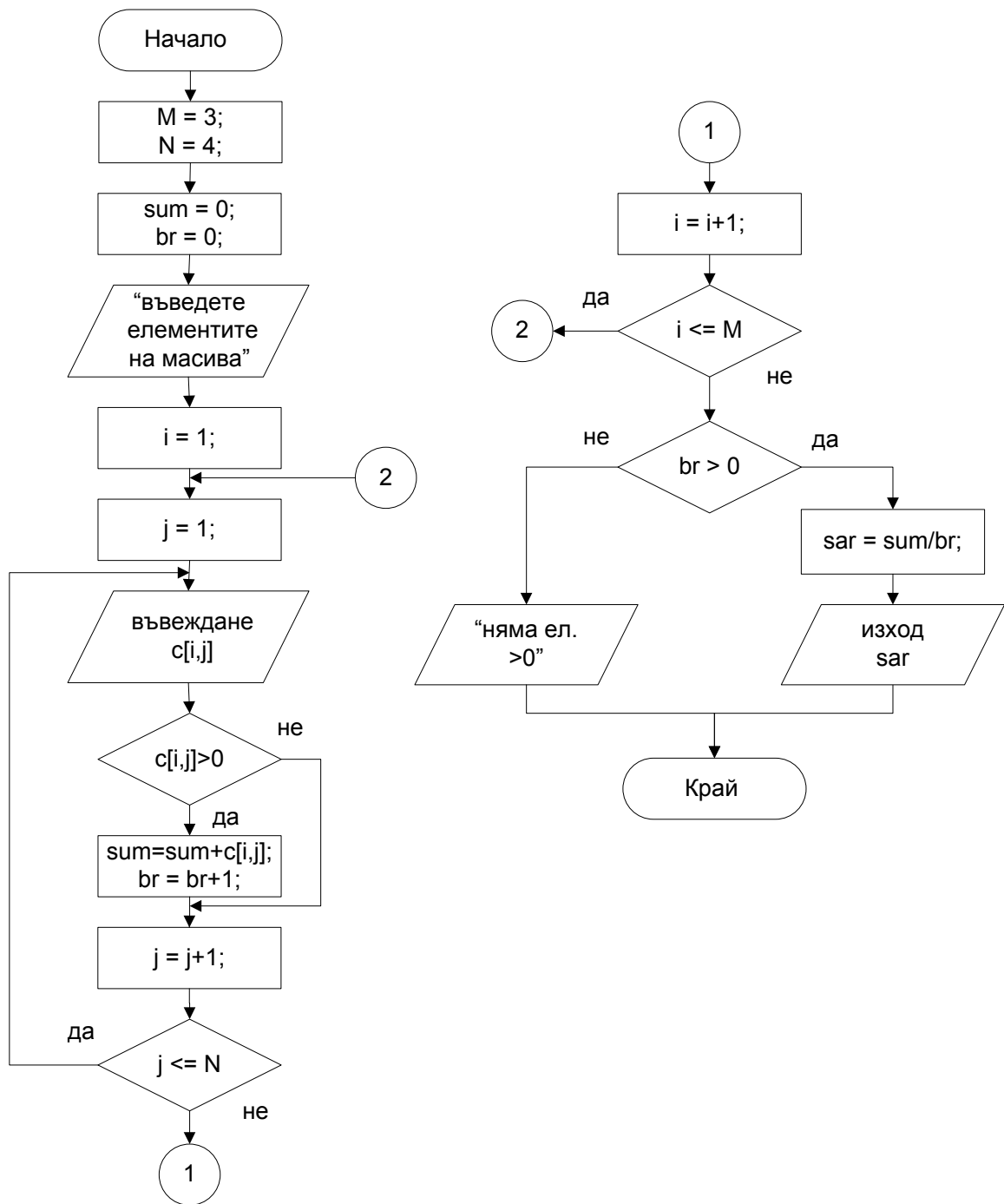
sum – сума на положителните елементи от масива;

br – брой на положителните елементи от масива;

sar – средноаритметична стойност.

1) Програма на Pascal

```
Program e_1;
const M=3; N=4;
type MasC=Array[1..M,1..N] of integer;
Var C:MasC;
    br,sum,i,j:integer;
    sar:real;
Begin
    br:=0;
    sum:=0;
    Writeln('въведете елементи на матрица с ',M,'реда
            ',N,'стълба');
```



{ организиране на 2 вложени цикъла за двумерния масив:
въвеждане, проверка на елементите, сумиране и броене }

For i:=1 to M do

For j:=1 to N do

begin

write('c[' , i , ' , ' , j , ']=');

readln(c[i,j]);

if c[i,j]>0 then {проверка за елементи >0 }

begin

sum:=sum+c[i,j];

br:=br+1;

end;

end;

```

    if br>0 then begin      {проверка за налични елементи}
        sar:=sum/br;
        writeln('sar=',sar:0:2)
    end
    else      writeln('няма елементи >0');
readln;
End.

```

2) Програма на C++

```

#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{ const int  M=3,  N=4;    // брой редове и стълбове
  int  c[M][N];          // двумерен масив
  int  i, j, br=0, sum=0;
  double  sar;
  cout <<"Въведете елементи на матрица с " << M << "реда
    " << N << "стълба \n";
  /* организиране на 2 вложени цикъла за двумерния масив:
  въвеждане, проверка на елементите, сумиране и броене */
  for (i=0; i<M; i++)
    for (j=0; j<N; j++)      // индексите са от 0
      { cout << "c[" << i+1 << ", " << j+1 << "]=";
        cin >> c[i][j];
        if (c[i][j]>0)      // проверка за елементи >0
          {
              sum += c[i][j];
              br++;
          }
      }
  if (br>0)      // проверка за налични елементи
  {
  // пресмятане с явно преобразуване към реален тип:
    sar = (double)sum/br;
    cout << "За >0 средна ст.=" << sar << endl;
  }
  else
    cout << "Няма елементи >0 \n";
  system("PAUSE");
  return EXIT_SUCCESS;
}

```

Задача е_2: Да се състави програма по следното условие: Даден е двумерен масив $D(N,N)$ с произволни въведени реални стойности, като $N \leq 50$. Да се изведе матрицата в табличен вид на екрана. Да се намери произведението на елементите, които са по-големи от 10 и са разположени над главния диагонал. Да се изведе получения резултат.

Решение: При въвеждането на стойност за брой редове и стълбове е необходимо да се направи проверка за коректни данни. За извършване на проверката може да се използва оператор за цикъл с предусловие или следусловие. В случая е използван оператор със следусловие.

Следва въвеждане на стойности за елементите на двумерния масив, а после – извеждането му в табличен вид.

Обработката на масива над главния диагонал би могла да е с проверка за индексите да са $i < j$, или както е реализирано в тази задача индексите i и j в цикличните структури се изменят по подходящ начин и така масивът се обработва само над главния диагонал. Елементите от матрицата се проверяват по зададеното условие. Извежда се полученото произведение.

Използват се следните променливи:

d – име на двумерен масив (матрица);

n – въведен брой на редове и стълбове;

p – променлива за произведението;

i, j – управляващи променливи за цикли и текущи индекси на елемент от масива;

1) Програма на C

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

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

```
int main(int argc, char *argv[])
```

```
{ int i, j, n;
```

```
float d[50][50];
```

```
double p;
```

```
do
```

```
{ printf("Въведи брой за ред и стълб (n<=50) n=");
```

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

```
} while (n < 2 || n > 50);
```

```
printf("\n Въведи елементи на матрицата \n");
```

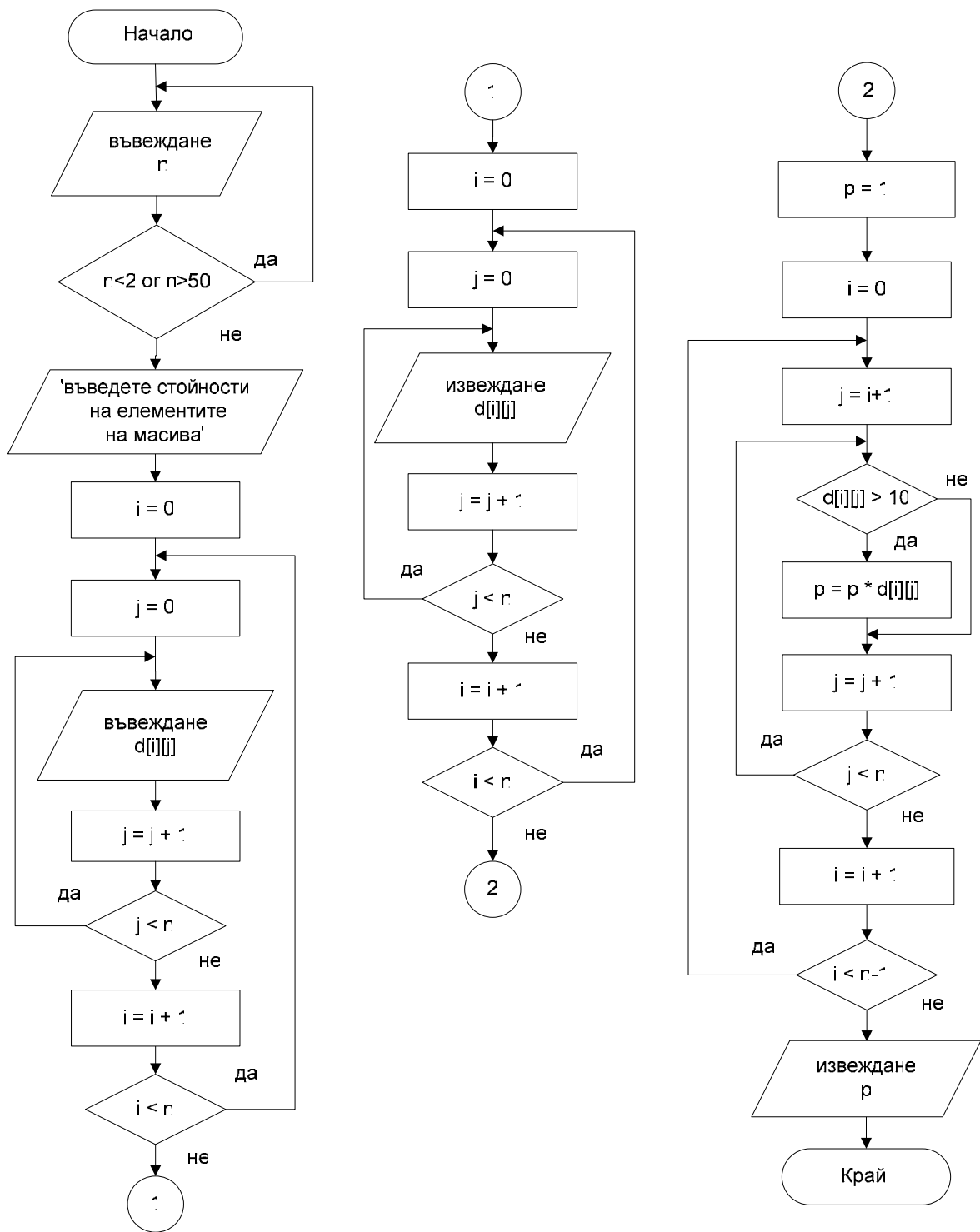
```
for (i=0; i<n; i++) // въвеждане на матрицата
```

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

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

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

```
}
```



```

// извеждане на матрицата в табличен вид
printf("\n\n\t Матрицата е: \n");
for (i=0; i<n; i++)
{ for (j=0; j<n; j++)
    printf(" %10.2f   ", d[i][j]);
  printf("\n");
}
p=1;
// обработка на матрицата
  
```

```

    for (i=0; i<n-1; i++) // обработка над гл.диагонал
        for (j=i+1; j<n; j++)
            if (d[i][j] > 10)
                p *= d[i][j];
    printf("\n Произведението на елементите >10 над
           главния диагонал p = %e \n", p);
    system("PAUSE");
    return 0;
}

```

2) Програма на C++

```

#include <cstdlib>
#include <iostream>
#include <iomanip>
using namespace std;

int main(int argc, char *argv[])
{
    int i, j, n;
    float d[50][50];
    double p=1;
    do
    {
        cout << "Въведи брой за ред и стълб (n<=50) n=";
        cin >> n;
    } while (n<2 || n>50);
    cout << "\n Въведи елементи на матрицата \n";
    for (i=0; i<n; i++) // въвеждане на матрицата
        for (j=0; j<n; j++)
            {
                cout << "d[" << i+1 << ", " << j+1 << "]=";
                cin >> d[i][j];
            }
    cout << "\n\n\t Матрицата е: \n";
    for (i=0; i<n; i++) // извеждане на матрицата
    {
        for (j=0; j<n; j++)
            cout << setw(10) << d[i][j] << " ";
        cout << endl;
    }
    for (i=0; i<n-1; i++) // обработка над гл.диагонал
        for (j=i+1; j<n; j++)
            if (d[i][j] > 10)
                p *= d[i][j];
    cout << "\n Произведението на елементите >10 над
           главния диагонал p = " << p << endl;
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

Задача е_3: Да се състави програма със следното условие: Даден е двумерен масив $A(4,5)$ с произволни целочислени стойности. Да се изведе матрицата в табличен вид. Да се намери сумата на отрицателните елементите по редове. Получените стойности да се въведат в едномерен масив. Да се определи в кой ред се съдържа минималната сума I да се изведе получения резултат.

Решение: Първата стъпка при решаването на задачата е да се въведат от клавиатурата стойности за елементите на масива. След това въведеният масив се извежда на екрана, за да се виждат кои са елементите в отделните му редове.

Както при въвеждането и извеждането, така и при обработката на двумерния масив, изменението на броячите за индекси i и j на елемент от масива най-често са за обработка по редове: във външният цикъл се определя кой ред i се обработва, а вътрешният цикъл изменя брояча j за елементите вътре в реда. В тази задача при обработката е необходимо да се прави проверка дали стойността на елемента е отрицателна, за да се намери сумата на отрицателните елементи.

Променливата за сумата **sum** трябва да започва с начална стойност равна на 0 за всеки отделен ред от масива. След приключване на обработката на реда, полученият резултат може да се изведе, но за да се съхранят всички пресметнати суми на отделните редове, стойността на **sum** трябва да се присвои на елемент от едномерен масив **s**.

След извеждане на получените суми в едномерния масив **s**, той се обработва като се намира минималния му елемент и неговия индекс.

Използват се следните променливи:

a – име на двумерен масив;

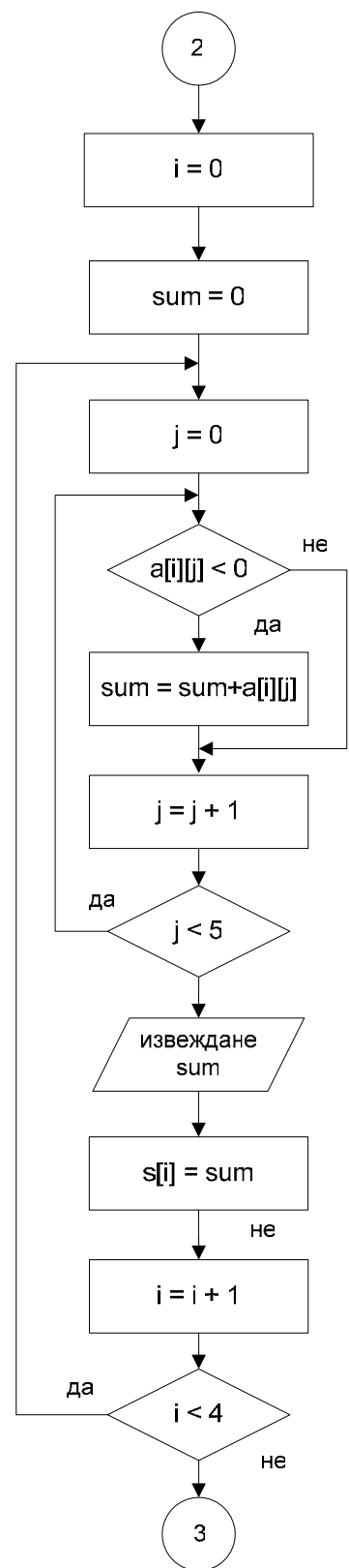
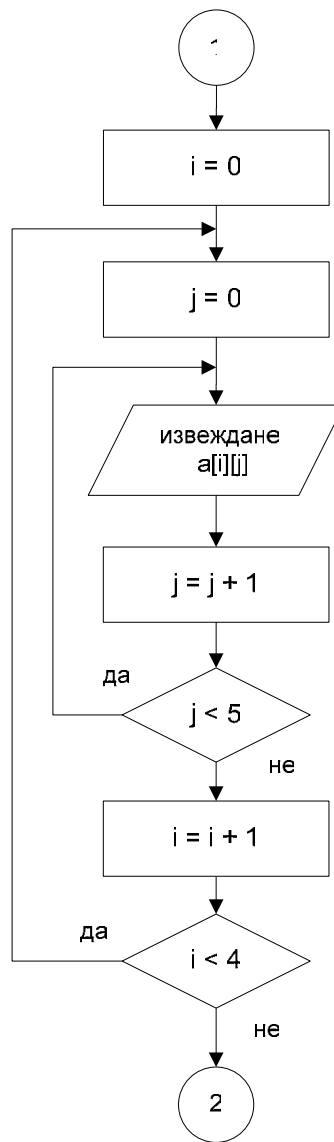
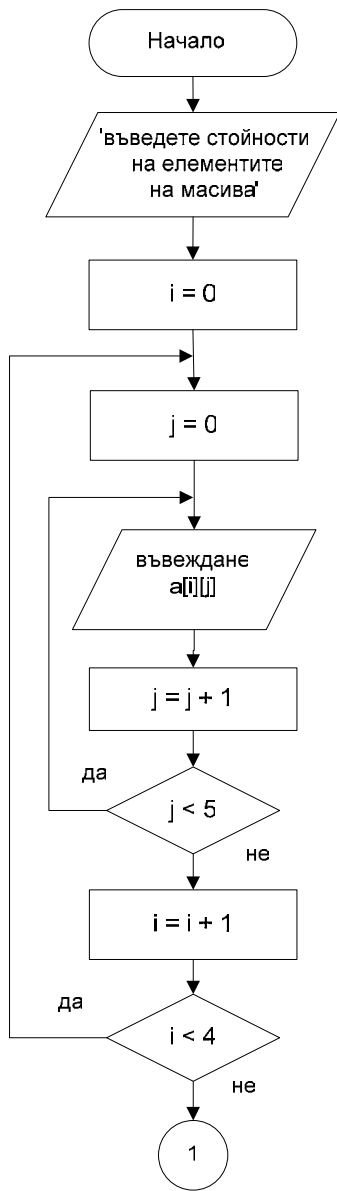
s – име на едномерен масив с резултатите за сумите от всеки ред;

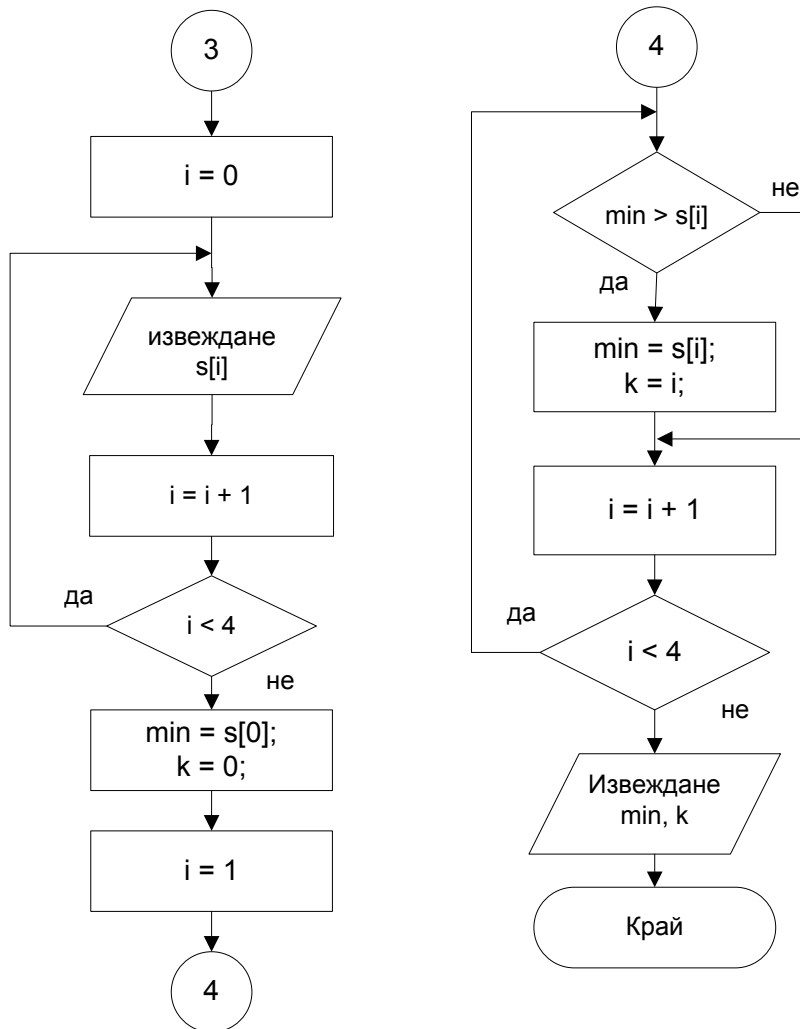
sum – сума на елементите на текущ ред от масива;

i, j – управляващи променливи за циклите и текущи индекси на елемент от масива;

min – променлива за минималната стойност от едномерния масив;

k – променлива за номера на минималния елемент от едномерния масив.





Програма на C++

```

#include <cstdlib>
#include <iostream>
#include <iomanip>

using namespace std;

int main(int argc, char *argv[])
{
    int a[4][5], s[4];
    int i, j, sum, min, k;
    // въвеждане на матрицата от клавиатурата
    cout << "Въведи елементи на матрицата A(4,5) \n";
    for (i=0; i<4; i++)
        for (j=0; j<5; j++)
            { cout << "a[" << i+1 << ", " << j+1 << "]=";
              cin >> a[i][j];
            }
}

```

```

// извеждане на матрицата на екрана
cout << "\n\n Матрицата A(4,5) е: \n";
for (i=0; i<4; i++)
{   for (j=0; j<5; j++)
        cout << setw(8) << a[i][j] << " ";
    cout << endl;
}
cout << endl;
// обработка на матрицата A(4,5) по редове
for (i=0; i<4; i++)
{   sum = 0;           // начална стойност за текущия ред
    for (j=0; j<5; j++) // обработка на текущия ред
        if (a[i][j]<0)
            sum += a[i][j];
    cout << "Сумата на " << i+1 << " ред е " << sum
        << endl;
    s[i] = sum;       // съхраняване в едномерен масив
}
// извеждане на едномерния масив s(4)
cout << "\n\n Масив s(4) със сумите на елементите по
        редове, по-малки от 0: \n";
for (i=0; i<4; i++)
    cout << setw(8) << s[i] << " ";
cout << endl;
// търсене на минимален елемент в едномерния масив s(4)
min = s[0];
k = 0;
for (i=0; i<4; i++) // обработка на масив s(4)
    if (min > s[i])
        {   min = s[i];
            k = i;
        }
cout << "\n Най-малката сума е:" << min;
cout << " на ред: " << k+1 << endl;
system("PAUSE");
return EXIT_SUCCESS;
}

```

Задача е_4: Да се състави програма със следното условие: Даден е двумерен масив $T(M,N)$ с произволни целочислени стойности, като $M \leq 20$ и $N \leq 10$. Стойностите да се създадат с използване на генератор за произволни числа. Да се намери броят на елементите, кратни на 4, за всеки стълб от масива. Получените стойности да се съхранят в едномерен масив. Да се определи стълбът, в който се съдържа максималния брой. Да се изведе резултатът.

Решение: При въвеждането на стойности за брой редове M и стълбове N е необходимо да се направи проверка за коректни данни. За извършване на проверката в този пример се използва оператор за цикъл с предусловие.

За получаване на стойности за елементите на масива T в тази задача е използван генератор за случайни числа. Първо генераторът се стартира и при език C/C++ това най-често е със стойност взета от текущото време, така се генерират всеки път различни числа.

Полученият двумерен масив T се отпечатва на екрана.

След това обработката на масива T е по стълбове. Това означава, че за изменение на индексите i и j във външната циклична структура се задава стълба j , а във вътрешния цикъл се изменя индекс i за елементите в текущия стълб. Елементите се проверяват по зададеното условие и пресметнатият брой за резултат в обработения стълб се присвоява в едномерен масив b .

Полученият едномерен масив b се извежда. Намира се и се извежда максималният му елемент и неговият индекс.

Използват се следните променливи:

t – име на двумерен масив;

b – име на едномерен масив с резултати по стълбове;

br – броят на елементите на масива;

i, j – управляващи променливи за цикли и текущи индекси на елемент на масива;

max – променлива за максималната стойност от едномерния масив;

k – променлива за номера на максималния елемент от едномерния масив;

1) Програма на Pascal

```
program e_4;
```

```
var
```

```
  t: array [1..20,1..10] of integer;
```

```
  b: array [1..10] of integer;
```

```

    i, j, k, m, n, br, max: integer;
begin
    {определяне броя на редовете}
    writeln ('въведете брой редове m=');    readln (m);
    while (m<=1) or (m>20) do
        begin
            writeln (' некоректни данни m<=20 ');
            writeln ('въведете отново m=');
            readln (m);
        end;
    {определяне броя на стълбовете}
    writeln ('въведете брой стълбове n=');    readln (m);
    while (n<=1) or (n>10) do
        begin
            writeln (' некоректни данни n<=10 ');
            writeln ('въведете отново n=');
            readln (n);
        end;
    {генериране на стойности за елементите на масива}
    Writeln;
    Writeln('генериране стойности за елементи на
            масива');
    randomize;    {старт на генератора на случайни числа}
    for i:=1 to m do
        for j:=1 to n do
            t[i,j]:=random(21)-10;
            {генерира се число и се присвоява на елемента}
    writeln;
    {извеждане на елементите в табличен вид}
    for i:=1 to m do
        begin
            for j:=1 to n do
                write(t[i,j]:3,' ');
            writeln;
        end;
    writeln;
    for j:=1 to n do    {обработка по стълбове}
        begin
            br:=0;    {нулиране на брояча}
            for i:=1 to m do
                if t[i,j] mod 4=0 then    {проверка на условието}
                    br:= br+1;    {определяне на броя елементи}
            writeln('брой на ',j,' стълб = ',br:3);
            b[j]:=br;    {въвеждане в едномерен масив}
        end;
end;

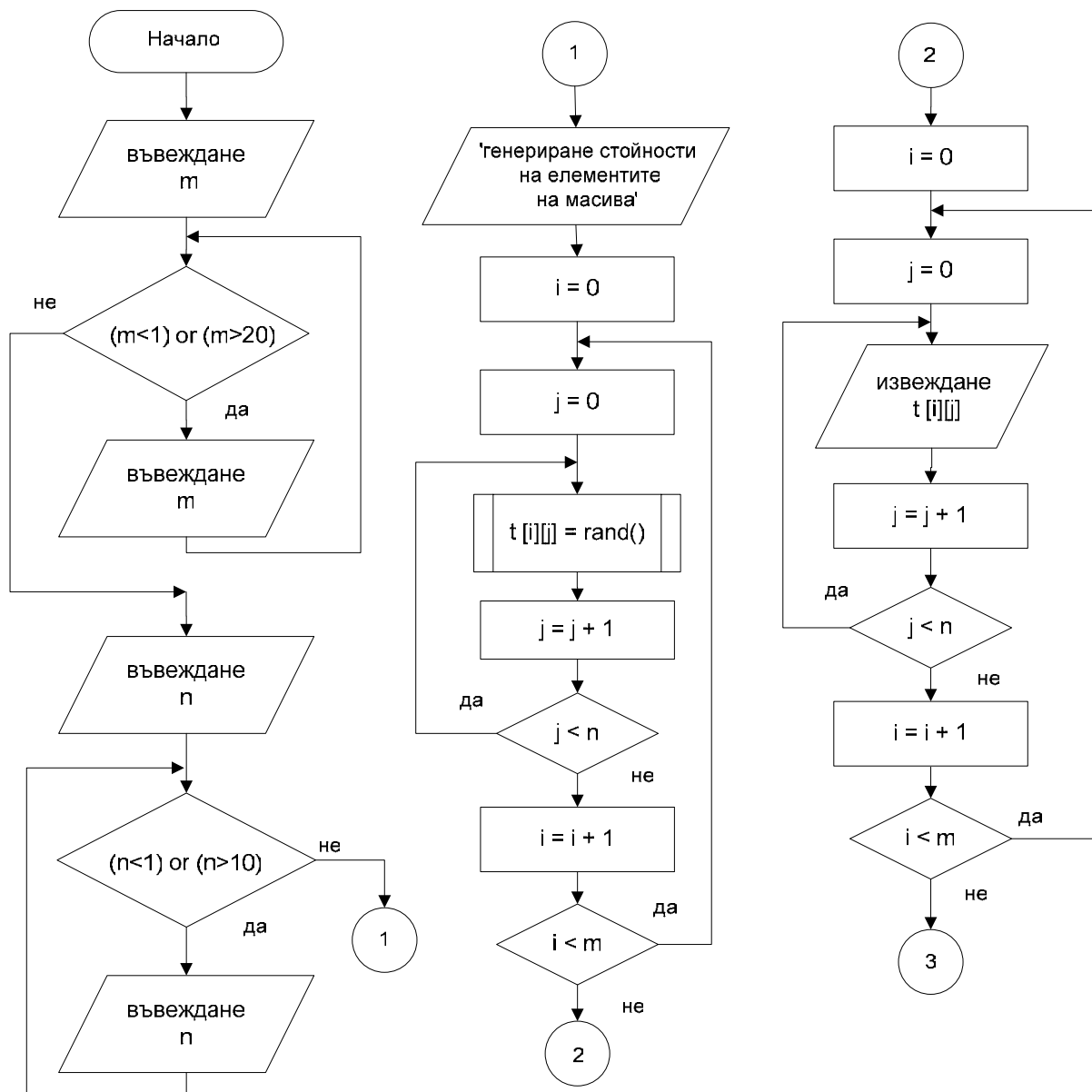
```

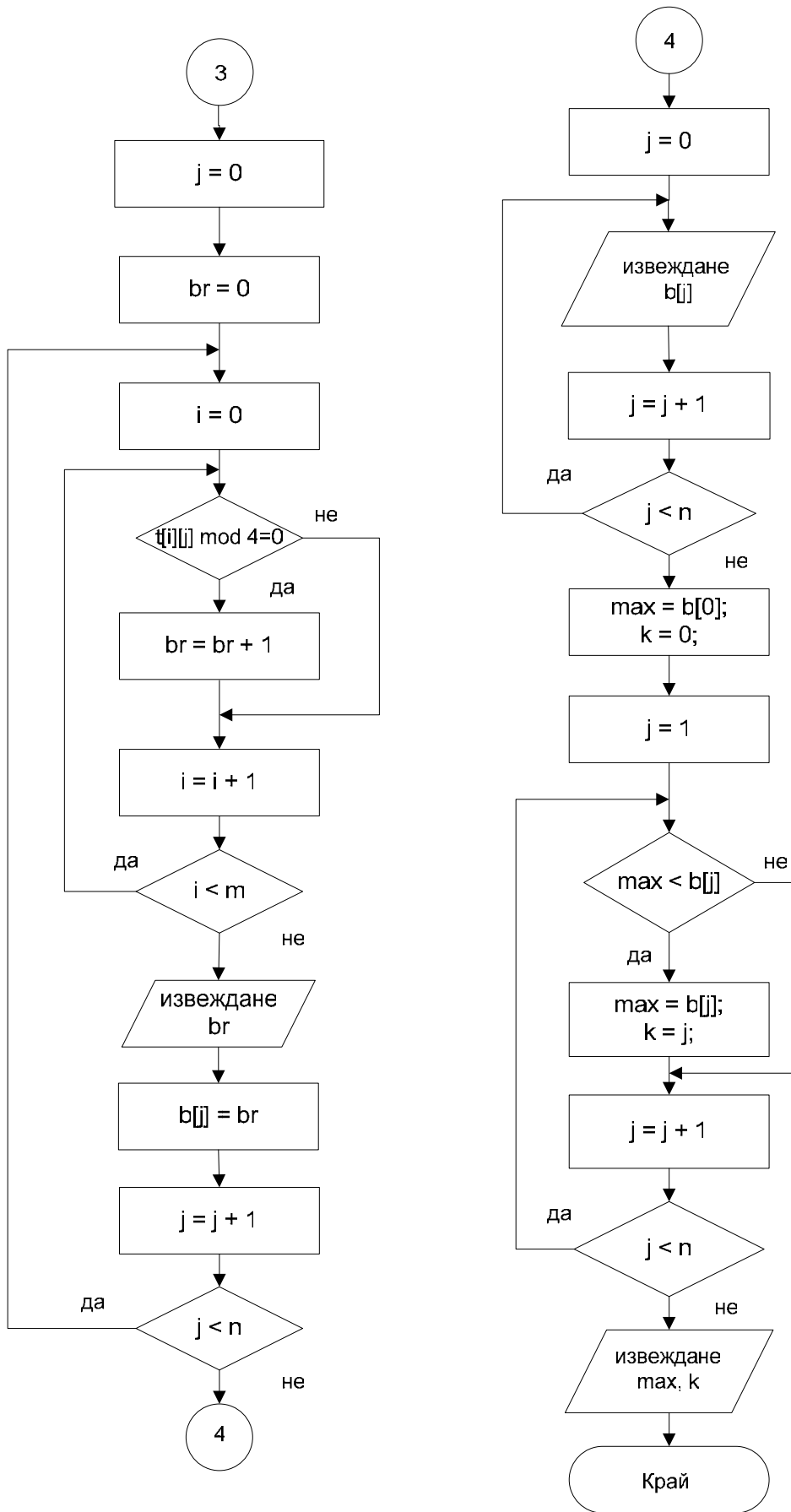
{извеждане на едномерния масив}

```
for j:= 1 to n do  
    write(b[j], ' ');  
writeln;
```

{намиране на максимален елемент в едномерен масив}

```
max:=b[1];  
k:=1;  
for j:= 2 to n do  
    if max < b[j] then  
        begin  
            max:= b[j]; k:=j;  
        end;  
writeln ('max= ',max, ' от ',k,'-ия стълб');  
readln;  
end.
```





2) Програма на C++

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <ctime>

using namespace std;

int main(int argc, char *argv[])
{   int  t[20][10], b[10];
    int  i, j, m, n, br, max, k;
    // Въвеждане на броя на редовете M
    cout << "\n Въведи брой редове M = ";
    cin >> m;
    while(m<1 || m>20)    // проверка за коректност за M
    {   cout << "Некоректни данни! 1<M<20 \n ";
        cout << "Въведи отново брой редове M=";
        cin >> m;
    }
    // Въвеждане на броя на стълбовете N
    cout << "\n Въведи брой стълбове N = ";
    cin >> n;
    while(n<1 || n>10)    // проверка за коректност за N
    {   cout << "Некоректни данни! 1<N<10 \n ";
        cout << "Въведи отново брой стълбове N=";
        cin >> n;
    }
    // генериране на стойности за масива T(M,N)
    cout << "\n Генериране на матрицата T(" << m << ", "
        << n << ") \n";
    // стартира се генератора със стойност - текущото време
    srand((unsigned)time( NULL ));
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
            t[i][j] = rand();    // генерира число
    cout << "\n\n\t Матрицата T(" << m << ", " << n <<
        "): \n";
    // извеждане на двумерния масив T(M,N) в табличен вид
    for (i=0; i<m; i++)
    {   for (j=0; j<n; j++)
            cout << setw(8) << t[i][j] << " ";
        cout << endl;
    }
    cout << endl;
```

```

// обработка на матрицата T(M,N) по стълбове
for (j=0; j<n; j++)
{   br = 0;           // начална стойност за текущия стълб
    for (i=0; i<m; i++) // обработка на текущия стълб
        if (t[i][j]%4 == 0)
            br++;
    cout << "Брой кратни на 4 за " << j+1 << " стълб
           e: " << br << endl;
    b[j] = br;        // съхраняване в едномерен масив
}
// извеждане на едномерния масив b(n)
cout << "\n\n Масив b(N) с брой на кратните на 4 по
        стълбове:\n";
for (j=0; j<n; j++)
    cout << setw(8) << b[j] << " ";
cout << endl;
// търсене на максимален елемент в едномерен масив b(n)
max = b[0];
k = 0;
for (j=1; j<n; j++) // обработка на масив b(n)
    if (max < b[j])
    {   max = b[j];
        k = j;
    }
cout << "\n Най-голям брой e:" << max << " на стълб:
        " << k+1 << endl;
system("PAUSE");
return EXIT_SUCCESS;
}

```

Идея: Тук числата се генерират автоматично. Можете да тествате със задаване на по-голям брой редове и стълбове, като по условие те са до $M \leq 20$ и $N \leq 10$.

Използването на генератор за числата е удобно за тестване и настройка на програми с много данни. В повечето реални програми обаче, е необходимо да се обработват не случайни числа, а конкретни измерени стойности, описващи предмети или явления. Затова обикновено данните се въвеждат от клавиатурата или се прочитат от файл.

Задачи за самостоятелна работа:

1. Даден е двумерен масив $A(4,3)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намери сумата на положителните елементи. Да се изведе резултатът.
2. Даден е двумерен масив $B(7,5)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намери броят на отрицателните елементи. Да се изведе резултатът.
3. Даден е двумерен масив $C(6,7)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намери произведението и броя на елементите равни на 5.0. Да се изведе резултатът.
4. Даден е двумерен масив $C(8,9)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намери средноаритметичното на елементите. Да се изведе резултатът.
5. Даден е двумерен масив $D(5,10)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намери броят на положителните елементи по редове. Получените стойности да се съхранят в едномерен масив. Да се изведе полученият масив.
6. Даден е двумерен масив $E(n,n)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намери сумата на отрицателните елементи по стълбове. Получените стойности да се запишат в едномерен масив. Да се изведе полученият масив.
7. Даден е двумерен масив $D(m,m)$ с произволни реални стойности. Да се изведе матрицата в табличен вид. Да се намери произведението на ненулевите елементи по главния диагонал. Да се изведе резултатът.
8. Даден е двумерен масив $NM(m,m)$ с произволни реални стойности. Да се изведе матрицата в табличен вид. Да се намери произведението на елементите по-големи от 2.5, намиращи се над главния диагонал. Да се изведе резултатът.
9. Даден е двумерен масив $MN(4,4)$ с произволни реални стойности. Да се изведе матрицата в табличен вид. Да се намери броят на нулевите елементи, намиращи се под главния диагонал. Да се изведат резултатите.

10. Даден е двумерен масив $F(k,n)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намерят максималните елементи за отделните редове. Получените стойности да се съхранят в едномерен масив. Да се изведат полученият масив и резултатите.
11. Даден е двумерен масив $DI(n,k)$ с произволни реални стойности, да се изведе матрицата в табличен вид. Да се намерят минималните елементи за отделните стълбове. Получените стойности да се съхранят в едномерен масив. Да се изведат полученият масив и резултатите.
12. Даден е двумерен масив $KL(x,y)$ с произволни реални стойности. Да се изведе матрицата в табличен вид. Да се намери броят на елементите по-големи от 5 по редове. Получените стойности да се запишат в едномерен масив. Да се определи редът, в който се съдържа максималния брой. Да се изведат полученият масив и резултатите.
13. Даден е двумерен масив $KL(y,u)$ с произволни реални стойности. Да се изведе матрицата в табличен вид. Да се намери сумата на елементите по-малки от 10 по стълбове. Получените стойности да се запишат в едномерен масив. Да се определи стълбът, в който се съдържа минималната сума. Да се изведат получения масив и резултатите.
14. Даден е двумерен масив $TR(n,n)$ с произволни реални стойности. Да се изведе матрицата в табличен вид. Да се намери средноаритметичното на положителните елементи, намиращи се под главния диагонал. Да се изведе резултатът.
15. Даден е двумерен масив $AB(s,s)$ с произволни целочислени стойности. Да се изведе матрицата в табличен вид. Да се намери броят и произведението на елементите кратни на 3, намиращи се над главния диагонал. Да се изведе резултатът.
16. Даден е двумерен масив $ES(F,F)$ с произволни целочислени стойности. Да се изведе матрицата в табличен вид. Да се намери средноаритметичното на елементите кратни на 5, намиращи се върху главния диагонал. Да се изведе резултатът.
17. Даден е двумерен масив $DR(m,m)$ с произволни реални стойности. Да се изведе матрицата в табличен вид. Да се намери средноквадратичното на положителните елементи и средноаритметичното на отрицателните. Да се изведат резултатите.