

## ЛЕКЦІЯ 10

1. ДИНАМІЧНИЙ РОЗПОДІЛ ПАМ'ЯТІ
2. ПРОГРАМУВАННЯ РОБОТИ ФУНКЦІЙ З ДВОВИМІРНИМИ МАСИВИ У ПАМ'ЯТІ, ЩО ДИНАМІЧНО РОЗПОДІЛЯЄТЬСЯ

## Двовимірні масиви у пам'яті, що динамічно розподіляється

Для розташування двовимірних масивів використовується механізм «показчик на показчик»:

Показчик		Показчик		Змінна
адреса	→	адреса	→	Значення

```
<тип> ** pointer;
```

### ФУНКЦІЇ, ЩО ПРАЦЮЮТЬ З ДВОВИМІРНИМИ МАСИВАМИ

У випадку, коли необхідно використати масиви змінного розміру як параметри-змінні функцій, тобто при необхідності внесення змін в ці масиви або створення нових за допомогою інших, необхідно використовувати покажчики (для одновимірних масивів) або покажчики на покажчик (для двовимірних) як параметри функцій.

Розглянемо приклад, в якому програма складається з чотирьох функцій.

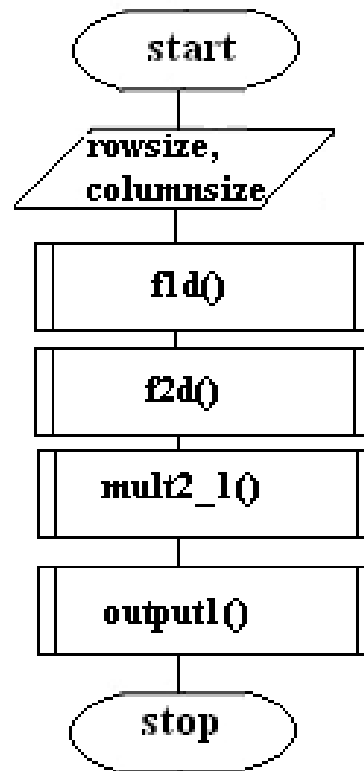
f1d() – задає значення одновимірному масиву;

f2d() – задає значення двовимірному масиву;

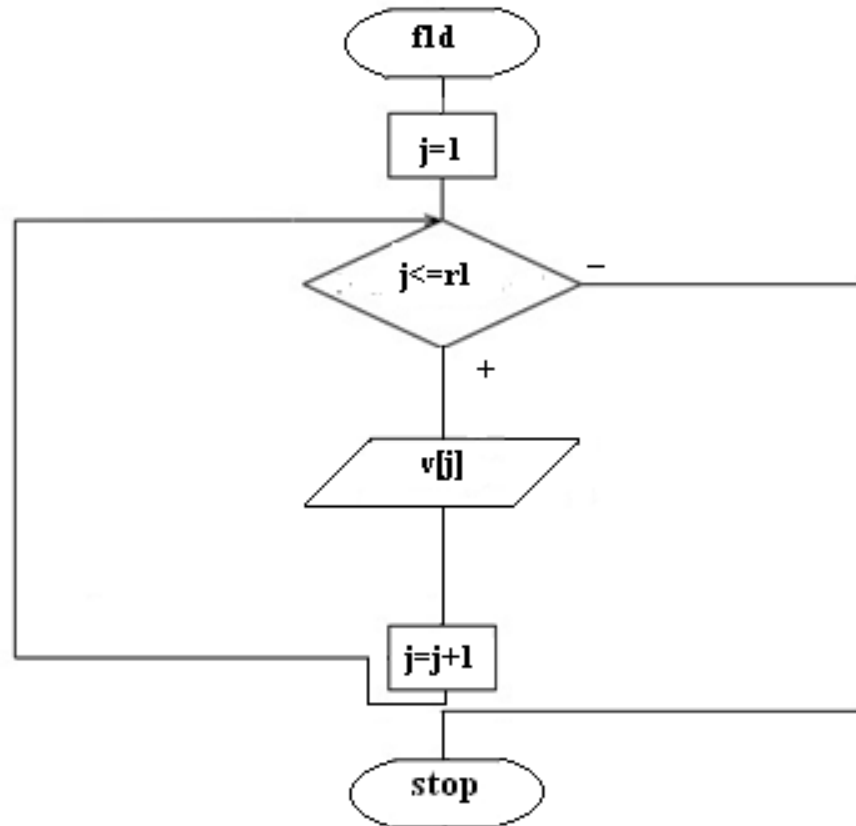
mult2\_1() – обчислює добуток матриці, значення компонентів якої задано у функції f2d(), на вектор, значення компонентів якого задано у функції f1d().

Результат виводиться на екран у функції output1();

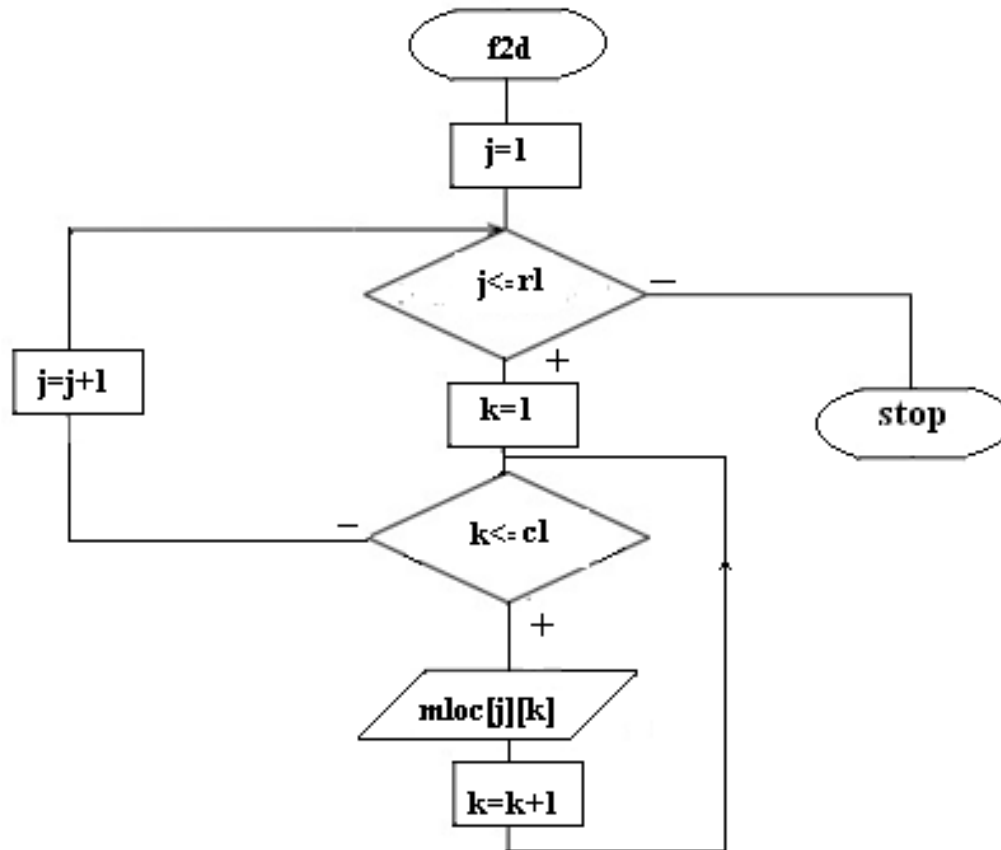
БЛОК-СХЕМА. ФУНКЦІЯ main()



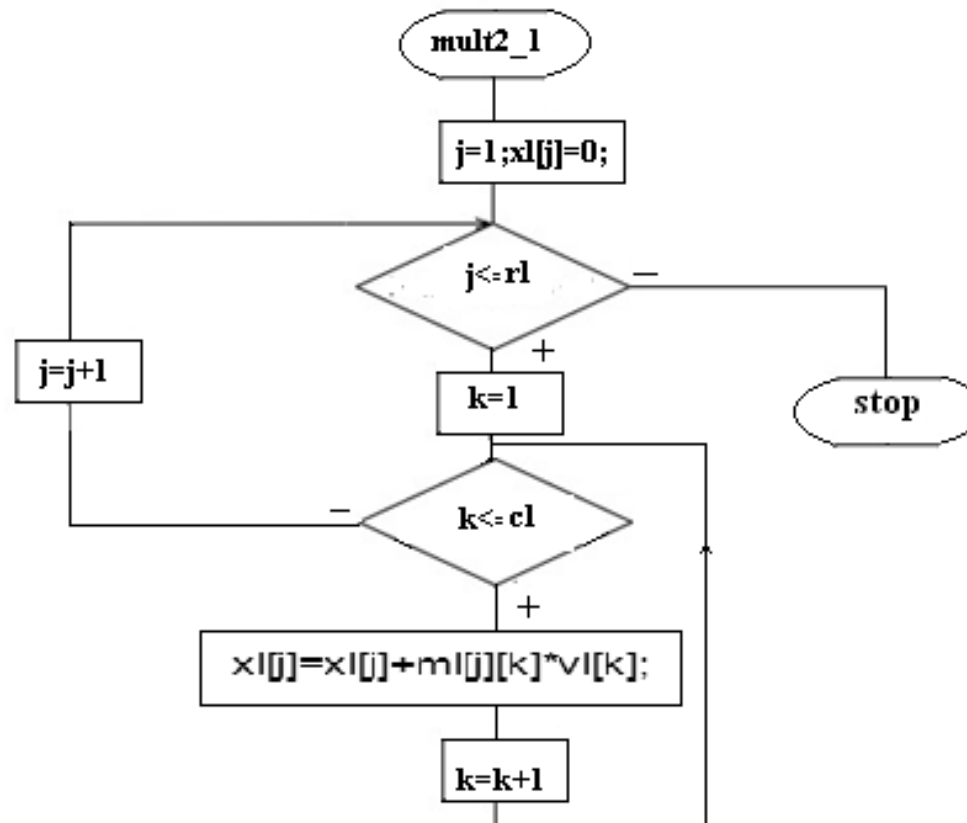
ФУНКЦІЯ f1d()



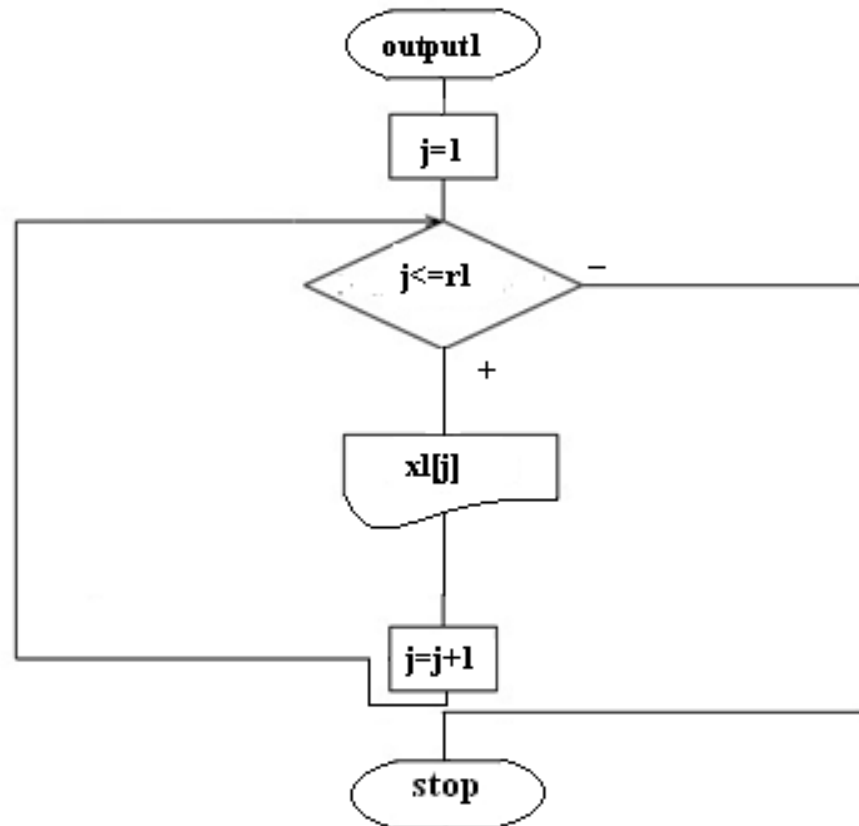
ФУНКЦІЯ f2d()



## ФУНКЦІЯ mult2\_1



ФУНКЦІЯ output1





```
*main.cpp X
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4  void f1d(double *, int);
5  void f2d(double **, int, int);
6  void mult2_1(double **, double*, double*, int, int);
7  void output1(double *, int);
8  int main()
9  {
10     int rowsize, columnsize, i, j;
11     double**m, *v,*x;
12     cout<<"Enter the number of rows: ";
13     cin>>rowsize;
14     cout<<"Enter the number of columns: ";
15     cin>>columnsize;
16     m=new double*[rowsize] ;
17     for (i=0;i<rowsize;i++)
18         m[i]= new double [columnsize];
19         v=new double[rowsize];
20         x=new double[rowsize];
```

```
21
22     f1d(v, rowsize-1);
23     f2d(m, rowsize-1, columnsize-1);
24     mult2_1(m, v, x, rowsize-1, columnsize-1);
25     /**for (i=1; i<rowsize; i++)
26     cout<<v[i]<<"\n";
27     for (i=1; i<rowsize; i++)
28     {
29         for (j=1; j<columnsize; j++)
30             cout<<m[i][j]<<" "; cout<<"\n";
31     };*/
32     output1(x, rowsize-1);
33
34     for (i=0; i<=rowsize; i++)
35         delete[] m[i];    delete[] m;
36         delete[] v; delete[] x;
37
38     return 0;
39 }
```

```
40 void f1d(double *vloc, int rowsizeloc)
41 {
42     int j;
43     ifstream in1 ("vector.txt");
44     for (j=1;j<=rowsizeloc;j++)
45         in1>>vloc[j];
46     in1.close();
47 }
48 void f2d(double **mloc, int rowsizeloc, int columnsizeloc)
49 {
50     int k,j;
51     ifstream in2 ("matrix.txt");
52     for (j=1;j<=rowsizeloc;j++)
53     for (k=1;k<=columnsizeloc;k++)
54         in2>>mloc[j][k];
55     in2.close();
56 }
```

```
57 void mult2_1(double **mloc, double*vloc, double*xloc,  
58             int rowsizeloc, int columnsizeloc)  
59 {  
60     int k,j;  
61     for (j=1;j<=rowsizeloc;j++)  
62     { xloc[j]=0;  
63       for (k=1;k<=columnsizeloc;k++)  
64         xloc[j]=xloc[j]+mloc[j][k]*vloc[k];  
65     }  
66 }  
67 void output1(double *xloc, int rowsizeloc)  
68 {  
69     int j;  
70     for (j=1;j<=rowsizeloc;j++)  
71         cout<<xloc[j]<<"\n";  
72 }  
73 }
```

# ПРОГРАМУВАННЯ

```
matrix.txt - Блокнот
Файл  Правка  Формат  Вид  Справка
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

```
vector.txt - Блокнот
Файл  Правка  Формат  Вид  Справка
1
1
1
1
```

```
C:\ "d:\CPP\C EDUCATION 1 sem 1 year\c91\
Enter the number of rows: 5
Enter the number of columns: 5
1
1
1
1
1
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
1
1
1
1
```

**ДЯКУЮ ЗА УВАГУ!**