

## ЛЕКЦІЯ 11

ПЕРЕДАЧА МАСИВІВ ДО ФУНКЦІЙ.  
ПРИКЛАДИ ПРОГРАМ

Виконання вимог структурно-логічного програмування обумовлює необхідність використання модульної структури програми, коли усі дії мають виконуватись у відповідних підпрограмах, що у мові C/C++ мають назву функцій.

Зв'язок між функціями забезпечується завдяки використанню у них параметрів-змінних (за визначенням, такі параметри можуть змінювати свої значення при поверненні у підпрограму, що їх викликала).

У мові C/C++ реалізацію параметрів-змінних виконано з використанням механізму покажчиків.

Дуже часто параметрами функцій є масиви. Покажчик на масив – його ідентифікатор. Масив, розміщений у пам'яті, що статично розподіляється, можливо передати до функції різними способами. Масив, який розташовано у пам'яті, що динамічно розподіляється, функція завжди приймає через відповідний покажчик.

У зв'язку з цим є відмінності у реалізації масивів як параметрів-змінних в залежності від їхнього створення – статичного або динамічного.

Продемонструємо це на прикладах.

Статична пам'ять. Вказуємо розмірність масиву.

```
1  #include <iostream>
2  using namespace std;
3  void function1(int arl[10]);
4  int main()
5  {int i, ar[10];
6   for (i=0;i<10;i++)
7     ar[i]=i*3;
8
9   function1(ar);
10  for (i=0;i<10;i++)
11    cout<<ar[i]<<"\n";
12    return 0;
13  }
14
15  void function1(int arl[10])
16  {
17      int j;
18      for (j=0;j<10;j++)
19          arl[j]=-arl[j];
20  }
```

Статична пам'ять. Використовуємо додатковий покажчик.

```
1  #include <iostream>
2  using namespace std;
3  void function1(int *);
4  int main()
5  {int i, ar[10], *p;
6   for (i=0; i<10; i++)
7     ar[i]=i*3;
8   p=&ar[0];
9   function1(p);
10  for (i=0; i<10; i++)
11    cout<<ar[i]<<"\n";
12    return 0;
13  }
14
15  void function1(int *ar1)
16  {
17    int j;
18    for(j=0; j<10; j++)
19      ar1[j]=-ar1[j];
20  }
```

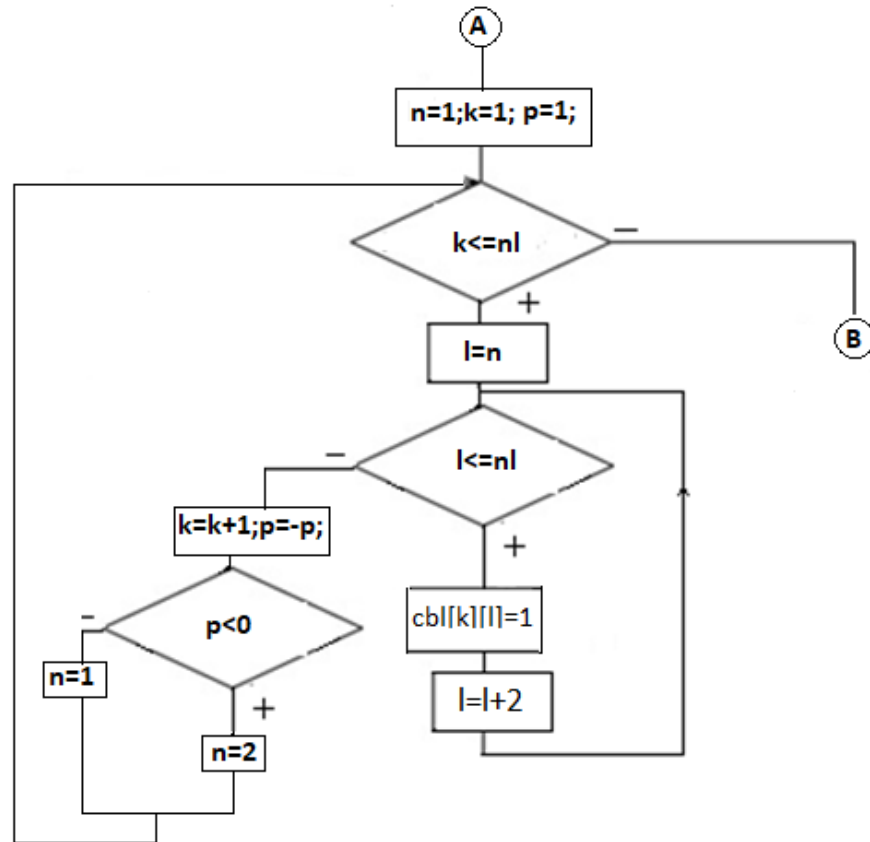
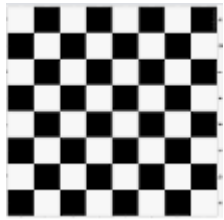
Статична пам'ять. Використовуємо покажчик на масив.

```
1  #include <iostream>
2  using namespace std;
3  void function1(int *);
4  int main()
5  {int i, ar[10];
6   for (i=0;i<10;i++)
7     ar[i]=i*3;
8
9   function1(ar);
10  for (i=0;i<10;i++)
11    cout<<ar[i]<<"\n";
12    return 0;
13  }
14
15  void function1(int *ar1)
16  {
17      int j;
18      for(j=0;j<10;j++)
19          ar1[j]=--ar1[j];
20  }
21
```

Динамічна пам'ять. Використовуємо покажчик на масив.

```
1  #include <iostream>
2  using namespace std;
3  void function1(int *);
4  int main()
5  {
6      int i, *ar;
7      ar= new int [10];
8      for (i=0;i<10;i++)
9          ar[i]=i*3;
10     function1(ar);
11     for (i=0;i<10;i++)
12         cout<<ar[i]<<"\n";
13     delete [] ar;
14     return 0;
15 }
16 void function1(int *ar1)
17 {
18     int j;
19     for(j=0;j<10;j++)
20         ar1[j]=-ar1[j];
21 }
```

П1. Розстановка комірок шахівниці. Блок-схема основного алгоритму.



## Програма розстановки комірок шахівниці

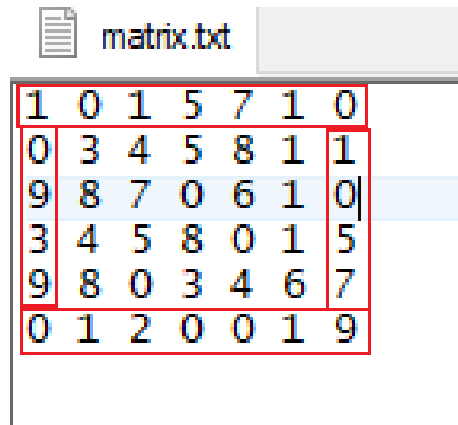
```
1  #include <iostream>
2  using namespace std;
3  void cells (int**, int );
4  int main()
5  {
6      int **chessboard;
7      int i,j,n;
8      cout<<"Enter the dimensions of chessboard (+1)"<<"\n";
9      cin>>n;
10     chessboard=new int*[n];
11     for(i=0;i<n;i++)
12         chessboard[i]=new int[n];
13
14     cells(chessboard,n-1);
15     for (i=1;i<=n-1;i++)
16     {for (j=1;j<=n-1;j++)
17         cout<<chessboard[i][j]<<" ";cout<<"\n";
18     }
19
20     for (i=0;i<n;i++)
21         delete[]chessboard[i];
22     delete[] chessboard;
23     return 0;
24 }
```



```
25 void cells (int** cbl, int nl)
26 {
27     int k,l,n,p=1;
28     for (k=1;k<=nl;k++)
29         for (l=1;l<=nl;l++)
30             cbl[k][l]=0;
31     n=1;k=1;
32     while (k<=nl)
33     {
34         for (l=n;l<=nl;l=l+2)
35             cbl[k][l]=1;
36         k++;
37         p=-p;
38         if(p<0) n=2; else n=1;
39     }
40
41 }
```

```
Enter the dimensions of chessboard (+1)
9
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1
```

П2. Визначити кількість нулів у граничних елементах двовимірного масиву



1	0	1	5	7	1	0
0	3	4	5	8	1	1
9	8	7	0	6	1	0
3	4	5	8	0	1	5
9	8	0	3	4	6	7
0	1	2	0	0	1	9

# ПРОГРАМУВАННЯ

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4  int zeroinframe(int**, int , int);
5  int main()
6  {
7      int **ar2;
8      int i,j,n,m;
9      cout<<"Enter the number of rows "<<"\n";
10     cin>>n;
11     cout<<"Enter the number of columns "<<"\n";
12     cin>>m;
13
14     ar2=new int*[n];
15     for(i=0;i<n;i++)
16         ar2[i]=new int[m];
17     ifstream in("matrix.txt");
18     for (i=0;i<n;i++)
19         for (j=0;j<m;j++)
20             in>>ar2[i][j];
21     in.close();
22     cout<<"Numbers of zeroes in array frame: "<<zeroinframe (ar2, n, m);
23     for (i=0;i<n;i++)
24         delete[] ar2[i];
25     delete[] ar2;
26     return 0;
27 }
```

```
28 int zeroinframe(int**ar2l, int nl , int ml)
29 {
30     int k,l,cnt=0;
31     for (k=0;k<nl;k=k+nl-1)
32         for (l=0;l<ml;l++)
33             if (ar2l[k][l]==0) cnt++;
34
35     for (k=1;k<nl-1;k++)
36         for (l=0;l<ml;l=l+ml-1)
37             if (ar2l[k][l]==0) cnt++;
38
39     return cnt;
40
41 }
```

ДЯКУЮ ЗА УВАГУ!