

## **Лабораторна робота**

### **Освоєння технології та принципів програмування роботи з двовимірними масивами на C++**

#### **Мета роботи**

**Освоїти технології та принципи програмування роботи з двовимірними масивами та отримати практичні навички виконання дій з їх елементами на C++.**

#### **Хід роботи**

- 1) Ознайомитись з методичними вказівками до лабораторної роботи та повторити тему "Масиви";
- 2) відповідно до завдання розробити схему алгоритму програми, що виконує зчитування чи заповнення масиву за вказаною схемою;
- 3) за схемою алгоритму написати програму на мові C++;
- 4) зробити висновки;
- 5) підготувати звіт про виконання лабораторної роботи, що включає наступні пункти: номер, тема, мета та хід лабораторної роботи, завдання, схема алгоритму програми, лістинг програми з коментуванням кожної інструкції, результат роботи програми (скріншот), висновки.

#### **Теоретичні відомості та рекомендації до виконання**

Людина завжди прагне до спрощення сприйняття навколишнього світу. Одним з проявів є сортування об'єктів і об'єднання їх у групи за будь-якою ознакою. Таким самим чином роблять і в програмуванні, об'єднуючи змінні з однаковим типом даних під одним ім'ям і користуючись ними як елементами групи. Це спрощує задачу сприйняття та обробки безлічі змінних одного типу даних. Такі групи називають *масивами*. Масиви можуть слугувати як для зберігання N-ї кількості значень, так і для виконання з ними операцій. Причому використання масивів не обмежується роботою з послідовністю значень. Існують також поняття багатовимірних масивів, які являють собою послідовності з послідовностей значень, що досить часто може бути зручно. В основному використовуються одновимірні, двовимірні та тривимірні масиви.

При виконанні завдання до лабораторної роботи студент повинен ознайомитися з особливостями оголошення та ініціалізації двовимірних масивів в мові C++, а також навчитися виконувати операції над ними.

Виконання лабораторної роботи починається ознайомленням з теоретичними відомостями про масиви. Потім студент приступає до розробки схеми алгоритму програми. Залежно від заданого викладачем варіанту завдання, виконується читання/запис значень; двовимірний масив розмірністю 5x5 значеннями згідно з табл. 5.9, та заповнення елементів масиву з "0" та "1" за алгоритмом, що зазначено в табл. 5.10 та виведення на друк. За схемою алгоритму здійснюється написання програми мовою C++. Всі схеми, наведені в таблицях, формуються на основі

формул залежностей між індексами за розмірностями. Так, для кожної з діагоналей двовимірного масиву

для правих діагоналей

	$i+j=0$	$i+j=1$	$i+j=2$	$i+j=3$	$i+j=4$
$i/j$	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>0</b>	0,0	0,1	0,2	0,3	0,4
<b>1</b>	1,0	1,1	1,2	1,3	1,4
<b>2</b>	2,0	2,1	2,2	2,3	2,4
<b>3</b>	3,0	3,1	3,2	3,3	3,4
<b>4</b>	4,0	4,1	4,2	4,3	4,4
					$i+j=5$
					$i+j=6$
					$i+j=7$
					$i+j=8,$

для лівих діагоналей

$i/j$	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	
<b>0</b>	0,0	0,1	0,2	0,3	0,4	$i-j=-4$
<b>1</b>	1,0	1,1	1,2	1,3	1,4	$i-j=-3$
<b>2</b>	2,0	2,1	2,2	2,3	2,4	$i-j=-2$
<b>3</b>	3,0	3,1	3,2	3,3	3,4	$i-j=-1$
<b>4</b>	4,0	4,1	4,2	4,3	4,4	
	$i-j=4$	$i-j=3$	$i-j=2$	$i-j=1$		$i-j=0.$

для правих діагоналей

	$i+j=0$	$i+j=1$	$i+j=2$	$i+j=3$	$i+j=4$
$i/j$	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>0</b>	0,0	0,1	0,2	0,3	0,4
<b>1</b>	1,0	1,1	1,2	1,3	1,4
<b>2</b>	2,0	2,1	2,2	2,3	2,4
<b>3</b>	3,0	3,1	3,2	3,3	3,4
<b>4</b>	4,0	4,1	4,2	4,3	4,4
					$i+j=5$
					$i+j=6$
					$i+j=7$
					$i+j=8,$

для лівих діагоналей

i/j	0	1	2	3	4	
0	0,0	0,1	0,2	0,3	0,4	i-j=-4
1	1,0	1,1	1,2	1,3	1,4	i-j=-3
2	2,0	2,1	2,2	2,3	2,4	i-j=-2
3	3,0	3,1	3,2	3,3	3,4	i-j=-1
4	4,0	4,1	4,2	4,3	4,4	
	i-j=4	i-j=3	i-j=2	i-j=1		i-j=0.

При правильному виборі алгоритму заповнення масиву програма дозволить виконувати заповнення масиву будь-якої розмірності!

У висновках до звіту студент повинен розкрити призначення теми " Принципи роботи з двовимірними масивами". Висновок має бути поданий в формі пояснення сфери застосування вивченого матеріалу згідно із розумінням його студентом.

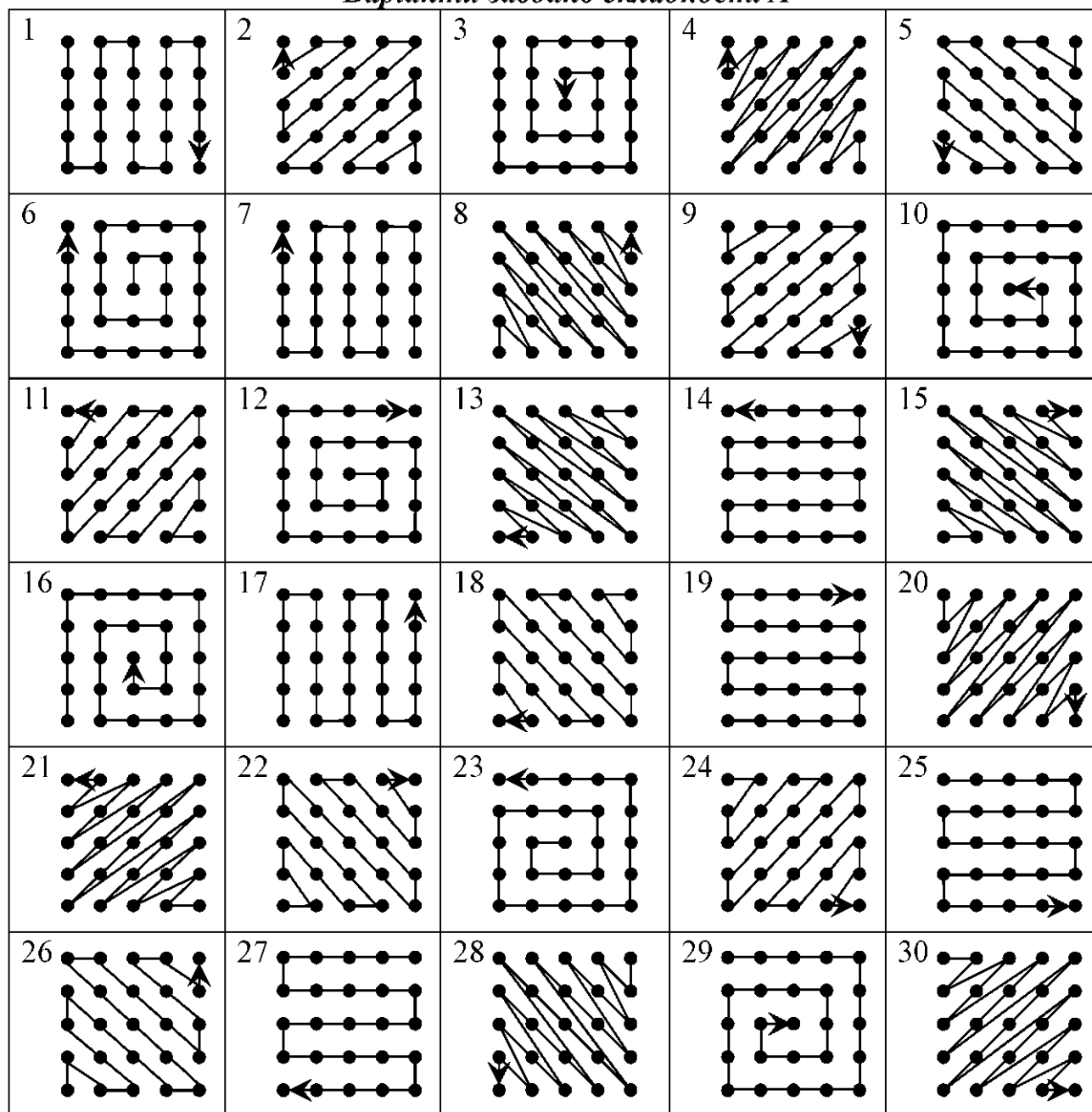
### Завдання для самостійного виконання

Розробити схему алгоритму та написати програму на мові C++, яка виконує (складність завдання обирається викладачем):

а) для парних варіантів читання, для непарних - запис значень елементів двовимірного масиву  $\text{Arr}[5,5]$  відповідно до вказаної у табл. 5.9 схеми. Результати як читання, так і запису повинні бути відображені на консолі. Значення елементів масиву приймаються від 1 до 25;

б) заповнення двовимірного масиву  $\text{Arr}[5,5]$  значеннями "0" та "1" згідно з зазначеною у табл. 5.10 схемою і вивід отриманої матриці на консоль.

## Варіанти завдань складності А



Таблиця 5.10

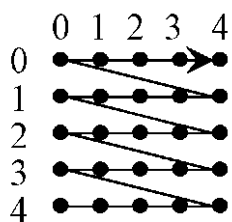
## Варіанти завдань складності Б

1	2	3	4	5																																																																																																																													
<table border="1"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1	1	0	0	0	1	0	0	0	0	<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table>	0	0	1	0	0	0	1	1	1	0	1	1	1	1	1	0	1	1	1	0	0	0	1	0	0	<table border="1"> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> </table>	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	<table border="1"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	1	1	1	0	0	0	1	1	0	1	0	1	1	0	0	0	1	1	1	1	1	1	<table border="1"> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> </table>	0	1	1	0	1	1	1	0	1	1	1	0	1	1	0	0	1	1	0	1	1	1	0	1	1
1	1	1	1	1																																																																																																																													
1	1	1	1	0																																																																																																																													
1	1	1	0	0																																																																																																																													
1	1	0	0	0																																																																																																																													
1	0	0	0	0																																																																																																																													
0	0	1	0	0																																																																																																																													
0	1	1	1	0																																																																																																																													
1	1	1	1	1																																																																																																																													
0	1	1	1	0																																																																																																																													
0	0	1	0	0																																																																																																																													
1	0	1	0	1																																																																																																																													
0	1	0	1	0																																																																																																																													
1	0	1	0	1																																																																																																																													
0	1	0	1	0																																																																																																																													
1	0	1	0	1																																																																																																																													
1	1	1	1	1																																																																																																																													
1	0	0	0	1																																																																																																																													
1	0	1	0	1																																																																																																																													
1	0	0	0	1																																																																																																																													
1	1	1	1	1																																																																																																																													
0	1	1	0	1																																																																																																																													
1	1	0	1	1																																																																																																																													
1	0	1	1	0																																																																																																																													
0	1	1	0	1																																																																																																																													
1	1	0	1	1																																																																																																																													

6	7	8	9	10
1 0 1 1 0	0 0 0 0 1	1 1 0 1 1	0 1 0 1 0	0 0 0 0 0
1 1 0 1 1	0 0 0 1 1	1 0 0 0 1	1 0 1 0 1	0 1 1 1 0
0 1 1 0 1	0 0 1 1 1	0 0 0 0 0	0 1 0 1 0	0 1 0 1 0
1 0 1 1 0	0 1 1 1 1	1 0 0 0 1	1 0 1 0 1	0 1 1 1 0
1 1 0 1 1	1 1 1 1 1	1 1 0 1 1	0 1 0 1 0	0 0 0 0 0
11	12	13	14	15
1 0 0 1 0	1 0 0 0 0	1 1 1 1 1	1 1 1 0 0	1 1 1 1 1
0 0 1 0 0	0 1 0 0 0	0 1 1 1 1	1 1 0 0 0	0 0 0 0 0
0 1 0 0 1	0 0 1 0 0	0 0 1 1 1	1 0 0 0 1	1 1 1 1 1
1 0 0 1 0	0 0 0 1 0	0 0 0 1 1	0 0 0 1 1	0 0 0 0 0
0 0 1 0 0	0 0 0 0 1	0 0 0 0 1	0 0 1 1 1	1 1 1 1 1
16	17	18	19	20
0 0 0 0 0	0 1 0 0 1	0 0 0 0 1	1 0 0 0 0	0 0 1 1 1
1 1 1 1 1	0 0 1 0 0	0 0 0 1 0	1 1 0 0 0	0 0 0 1 1
0 0 0 0 0	1 0 0 1 0	0 0 1 0 0	1 1 1 0 0	1 0 0 0 1
1 1 1 1 1	0 1 0 0 1	0 1 0 0 0	1 1 1 1 0	1 1 0 0 0
0 0 0 0 0	0 0 1 0 0	1 0 0 0 0	1 1 1 1 1	1 1 1 0 0
21	22	23	24	25
1 1 0 0 0	0 1 1 1 0	0 0 1 0 0	0 1 1 1 1	0 1 1 1 1
1 1 1 0 0	1 0 1 0 1	1 0 0 1 0	1 0 1 1 1	0 0 1 1 1
0 1 1 1 0	1 1 0 1 1	0 1 0 0 1	1 1 0 1 1	0 0 0 1 1
0 0 1 1 1	1 0 1 0 1	0 0 1 0 0	1 1 1 0 1	0 0 0 0 1
0 0 0 1 1	0 1 1 1 0	1 0 0 1 0	1 1 1 1 0	0 0 0 0 0
26	27	28	29	30
0 0 0 0 0	0 0 0 1 1	1 0 0 0 1	0 0 1 0 0	1 1 1 1 0
0 0 0 0 1	0 0 1 1 1	0 1 0 1 0	0 1 0 0 1	1 1 1 0 1
0 0 0 1 1	0 1 1 1 0	0 0 1 0 0	1 0 0 1 0	1 1 0 1 1
0 0 1 1 1	1 1 1 0 0	0 1 0 1 0	0 0 1 0 0	1 0 1 1 1
0 1 1 1 1	1 1 0 0 0	1 0 0 0 1	0 1 0 0 1	0 1 1 1 1

### Приклад виконання завдання

Складність А: необхідно заповнити двовимірний масив значеннями від 1 до 25 за наведеною нижче схемою і вивести отриману матрицю на консоль:



На схемі можна побачити, що при такому алгоритмі запису значень елементи кожного рядка перебираються від 0 до 4, а самі рядки від 4 до 0.

Виходячи з цього, розробимо схему алгоритму програми (рис. 5.9).

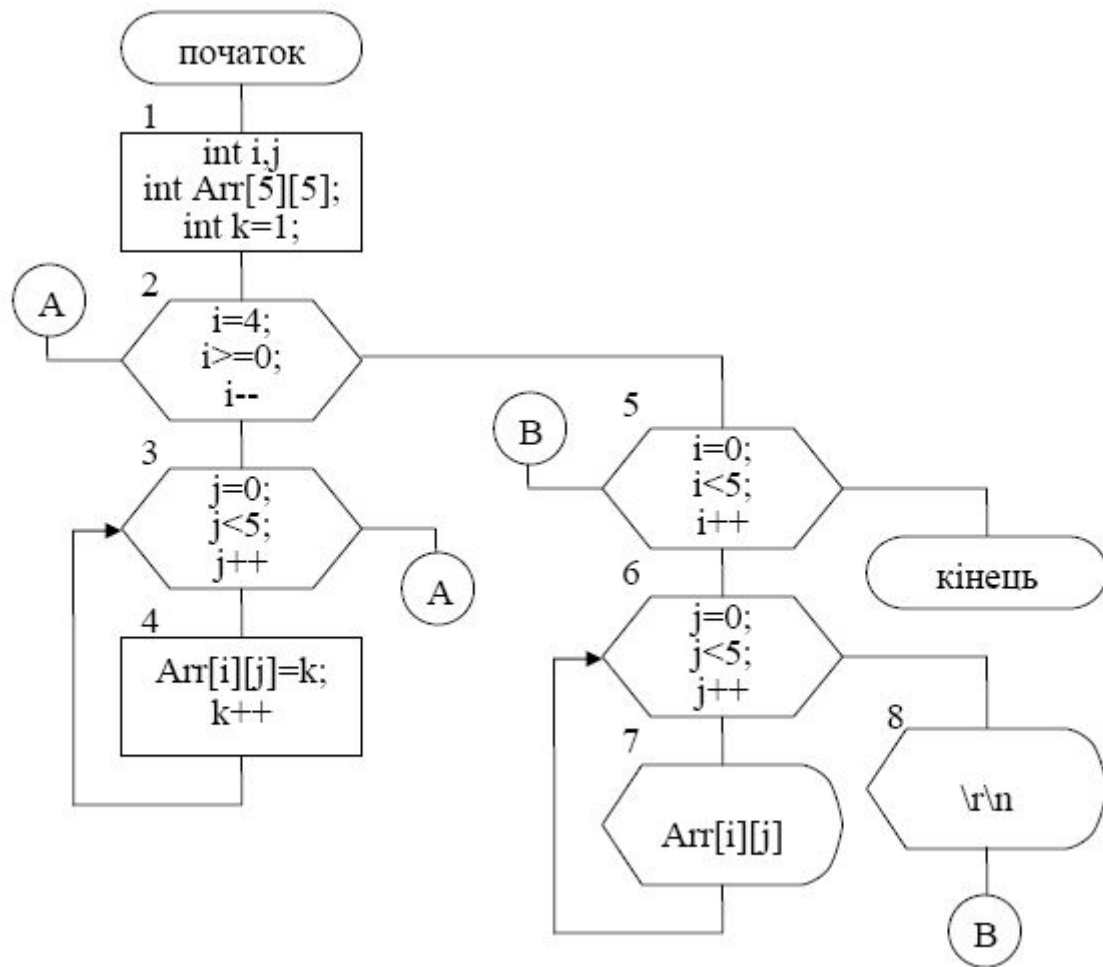


Рис. 5.9. Схема алгоритму програми, що заповнює двовимірний масив значеннями від 1 до 25

Згідно зі схемою алгоритму напишемо програму на мові C++:

```

#include <stdio.h> //підключення бібліотеки вводу/виводу
void main()       //оголошення та визначення головної функції
{
    int Arr[5][5]; //оголошення цілочисельного масиву з розмірністю 5x5
    int k=1;       //оголошення та визначення змінної лічильника
    int i, j;      //оголошення змінних лічильників для циклів
    for(i=4; i>=0; i--) //цикл перебору рядків у зворотному порядку
    {
        for(j=0; j<5; j++) //цикл перебору елементів рядка
        {
            Arr[i][j]=k; //визначення поточного елемента
            k++;          //збільшення лічильника
        }
    }
}

```

```

for(i=0; i<5; i++)           //цикл перебору рядків
{
    for(j=0; j<5; j++)         //цикл перебору елементів рядка
    {
        printf("%2i ", Arr[i][j]);    //вивід поточного значення
    }
    printf("\r\n");             //перехід на новий рядок
}
}

```

Результат виконання програми буде наступним

21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

Якщо в якості граничного значення для заповнення масиву використовувати змінну, що оголошена та визначена на початку програми, то алгоритм заповнення можна використовувати для правильного масиву будь-якої розмірності.

Змінимо програму, наведену вище, додавши змінну граничного значення та пов'язавши всі цикли з її значенням:

```

#include <stdio.h>             //підключення бібліотеки вводу/виводу
void main()                   //оголошення та визначення головної функції
{
    int n=7;                  //оголошення та визначення граничної змінної
    int Arr[7][7];           //оголошення цілочисельного масиву з розмірністю 7x7
    int k=1;                  //оголошення та визначення змінної лічильника
    int i, j;                 //оголошення змінних лічильників для циклів
    for(i=n-1; i>=0; i--)      //цикл перебору рядків у зворотному порядку
    {
        for(j=0; j<n; j++)     //цикл перебору елементів рядка
        {
            Arr[i][j]=k;       //визначення поточного елемента
            k++;               //збільшення лічильника
        }
    }
    for(i=0; i<n; i++)         //цикл перебору рядків
    {
        for(j=0; j<n; j++)     //цикл перебору елементів рядка
        {
            printf("%2i ", Arr[i][j]);    //вивід поточного значення
        }
        printf("\r\n");        //перехід на новий рядок
    }
}

```

Результат виконання програми буде наступним:

```

43  44  45  46  47  48  49
36  37  38  39  40  41  42
29  30  31  32  33  34  35
22  23  24  25  26  27  28
15  16  17  18  19  20  21
  8   9  10  11  12  13  14
  1   2   3   4   5   6   7

```

**Складність Б:** необхідно заповнити масив  $Arr[9][9]$  значеннями "0" та "1" за наступною схемою:

1	1	1	1	1
1	1	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

Як видно за схемою, значенням "1" заповнюються перший і останній стовпці, перший і останній рядки та діагоналі, всі інші елементи містять значення "0". Виходячи з цього, розробимо схему алгоритму роботи програми (рис. 5.10).

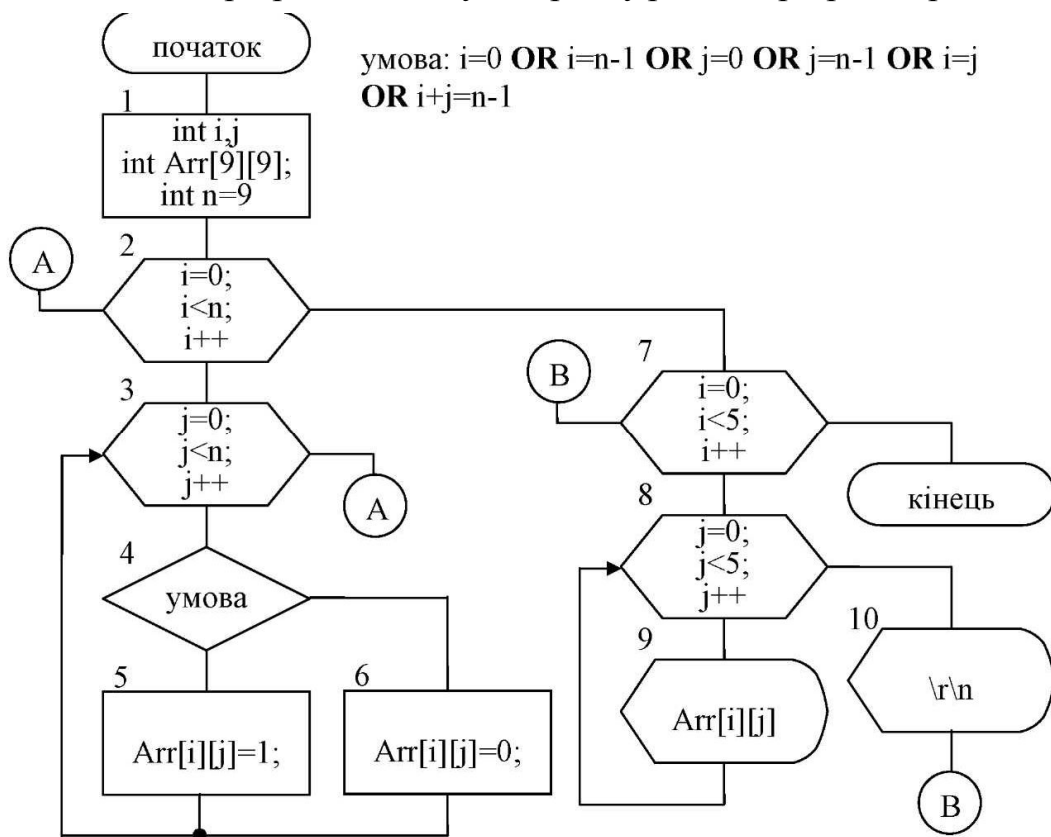


Рис. 5.10. Схема алгоритму програми, що заповнює двовимірний масив значеннями "0" та "1"



Згідно зі схемою алгоритму напишемо програму на мові C++:

```
#include<stdio.h>                                //підключення бібліотеки вводу/виводу

void main()                                     //оголошення та визначення головної функції
{
    int n=9;                                    //оголошення та визначення граничної змінної
    int Arr[9][9];                             //оголошення цілочисельного масиву з розмірністю 9x9
    int i, j;                                   //оголошення та визначення змінної лічильника
    for(i=0; i<n; i++)                          //цикл перебору рядків
        for(j=0; j<n; j++)                      //цикл перебору елементів рядка
//умова для запису значення "1"
            if(i==0 || i==n-1 || j==0 || j==n-1 || i==j || i+j==n-1)
                Arr[i][j]=1;                    //визначення поточного елементу
            else                                //інакше, запис значення "0"
                Arr[i][j]=0;                    //визначення поточного елемента
    for(i=0; i<n; i++)                          //цикл перебору рядків
        for(j=0; j<n; j++)                      //цикл перебору елементів рядка
            printf("%2i ", Arr[i][j]);          //вивід поточного елемента
            printf("\r\n");                     //перехід на новий рядок
}
```

Результат роботи програми буде наступним:

```
1 1 1 1 1 1 1 1 1
1 1 1 0 0 0 0 1 1
1 0 1 0 0 0 1 0 1
1 0 0 1 0 1 0 0 1
1 0 0 0 1 0 0 0 1
1 0 0 1 0 1 0 0 1
1 0 1 0 0 0 1 0 1
1 1 0 0 0 0 0 1 1
1 1 1 1 1 1 1 1 1
```

### Питання для підготовки до захисту лабораторної роботи

- 1) Які види багатовимірних масивів є найбільш поширеними?
- 2) Для чого застосовуються багатовимірні масиви?
- 3) За якими правилами можна пов'язувати елементи діагоналей матриці?
- 4) Як отримати доступ до елемента багатовимірного масиву?
- 5) Застосування яких операторів спрощує роботу з масивами?
- 6) Яка умова дозволяє вибірково заповнювати елементи правої діагоналі матриці?