

ЛАБОРАТОРНА РОБОТА №13

Використання динамічних числових масивів

Мета роботи: познайомитися з використанням динамічних масивів в мові програмування C.

Теоретичний вступ:

Відмінності *динамічного масиву* від звичайного полягають у тому, що:

- пам'ять під динамічний масив виділяється в процесі виконання програми за допомогою спеціальних функцій;
- кількість елементів динамічного масиву може бути задано змінною (але у програмі її неодмінно має бути визначено до виділення пам'яті під масив).

Синтаксис оголошення динамічного одновимірного масиву за допомогою функції `malloc()` є такий:

```
тип* ім'я = (тип*)malloc(sizeof(тип)*кількість_елементів);
```

Приклад оголошення дійсного динамічного масиву зі змінною кількістю елементів за допомогою функції `malloc()`:

```
int N = 10;  
float* a = (float*)malloc(sizeof(float) * N);
```

Приклад оголошення дійсного динамічного масиву зі змінною кількістю елементів за допомогою функції `calloc()`:

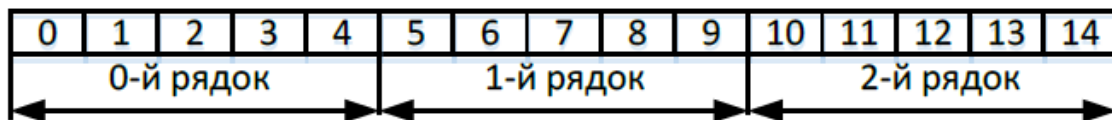
```
int N = 10;  
float* a = (float*)calloc(N, sizeof(float));
```

Звільнення пам'яті з-під цього масиву `a`:

```
free(a);
```

Існує можливість звертатися до елементів масиву без індексації за допомогою вказівників. У циклі за допомогою індексу поточний елемент масиву записується як `a[i]`. Згадаймо, що ім'я масиву `a` можна використовувати як вказівник на початок (нульовий елемент) масиву. Тоді, згідно з арифметикою вказівників, `a+i` — це вказівник на елемент, який міститься на `i` комірок далі від початку масиву, тобто вказівник на `a[i]`. Значення елемента `a[i]` можна записати за допомогою операції розадресації: `*(a+i)`.

Двовимірний динамічний масив з m рядків і n стовпчиків займає в пам'яті сусідні $m \cdot n$ комірок, тобто зберігається так само, як і одновимірний масив з $m \cdot n$ елементів. При розміщенні елементи двовимірних масивів розташовуються в пам'яті підряд один за одним з першого до останнього без проміжків у послідовно зростаючих адресах пам'яті. Наприклад, масив 3×5 зберігається у пам'яті в такий спосіб:



У такому масиві перші п'ять елементів належать до першого рядка, наступні п'ять – до другого і останні п'ять – до третього.

Нагадаємо, що a – вказівник на початок масиву, тобто на елемент $a[0][0]$. Щоб звернутися, наприклад, до елемента $a[1][3]$, слід “перестрибнути” від початку масиву через 5 елементів нульового рядка й 3 елементи першого рядка, тобто написати: $*(a+1*5+3)$. У загальному випадку до елемента $a[i][j]$ можна звернутися в такий спосіб: $*(a+i*5+j)$. Але цей спосіб роботи з двовимірним масивом є не надто зручний, тому що в програмі при звертанні до елемента масиву доводиться розадресовувати вказівник і обчислювати індекс елемента.

Оголосити дійсний динамічний масив 3×5 можна як одновимірний з 15-ти елементів:

```
float* a = (float*)malloc(3 * 5 * sizeof(float));
```

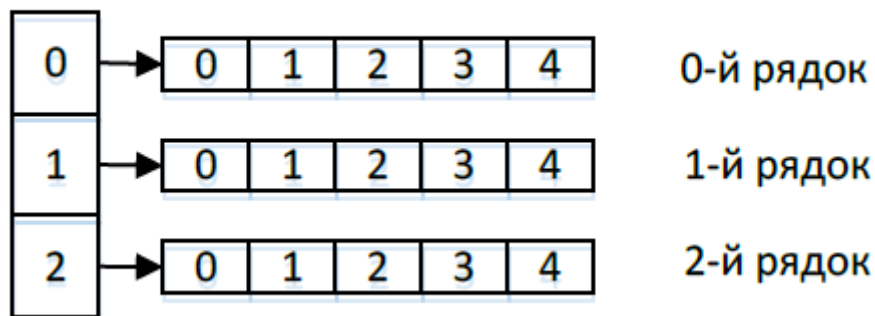
або

```
float* a = (float*)calloc(3 * 5, sizeof(float));
```

Пам'ять від створеного в такий спосіб масиву очищується за допомогою функції `free()`:

```
free(a);
```

Розглянемо інший спосіб роботи з динамічним двовимірним масивом. Для цього розмістимо в пам'яті матрицю 3×5 :



При цьому буде виділено пам'ять під кожний рядок матриці окремо, тобто буде утворено три різні одновимірні масиви. Адреси нульових елементів цих масивів зберігатимуться в допоміжному масиві `a` (пам'ять під нього слід виділити заздалегідь). Елементами цього масиву будуть адреси дійсних чисел, тому вони матимуть тип “вказівник на дійсне число”, тобто `float*`. Нагадаємо, що загальний вигляд оголошення вказівника на динамічний масив є такий:

тип елемента* ім'я масиву;

Тоді при оголошенні масиву `a` слід записати дві зірочки:

```
float** a = (float**)malloc(3 * sizeof(float*));
```

// Оголошення й розміщення в пам'яті допоміжного масиву з 3-х елементів типу `float*`

```
for (int i = 0; i < 3; i++)
```

```
  a[i] = (float*)malloc(5 * sizeof(float));
```

// У циклі виділяється пам'ять під 3 масиви по 5 елементів (рядки матриці)

// Значення адрес нульових елементів цих масивів записуються у відповідні елементи масиву `a`

Після цього можна працювати з матрицею як зі звичайним двовимірним масивом, звертаючись до кожного елемента за його індексом, наведеним у квадратних дужках: `a[i][j]`, – що є більш природним і зручним, аніж попередній спосіб.

Звільнення пам'яті необхідно виконувати в зворотному порядку – спочатку звільнити пам'ять під рядки в циклі, а потім звільнити пам'ять, що виділялась під масив вказівників.

```
for (int i = 0; i < 3; i++)
```

```
  free(a[i]);
```

```
free(a);
```

Аналогічним є оголошення за допомогою `calloc()`:

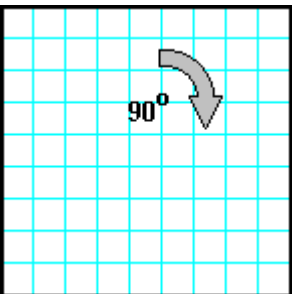
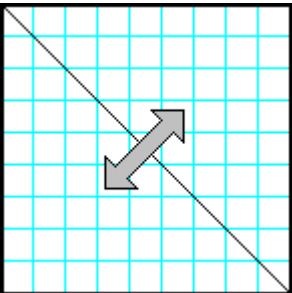
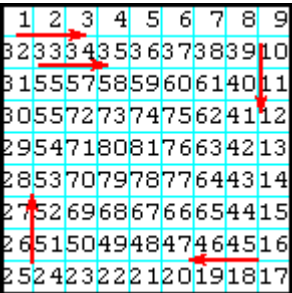
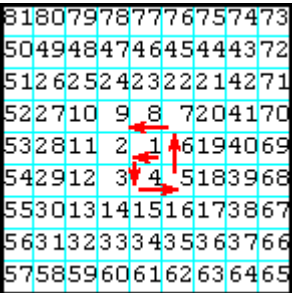
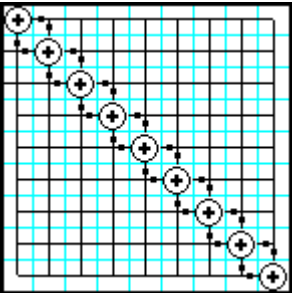
```
float** a = (float**)calloc(3, sizeof(float*));  
for (int i = 0; i < 3; i++)  
    a[i] = (float*)calloc(5, sizeof(float));
```

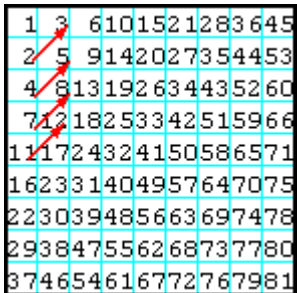
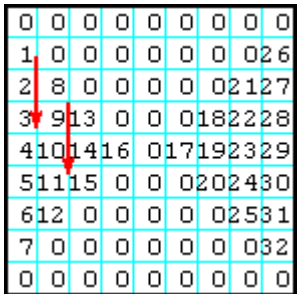
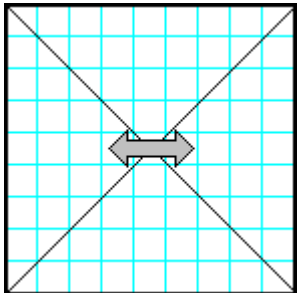
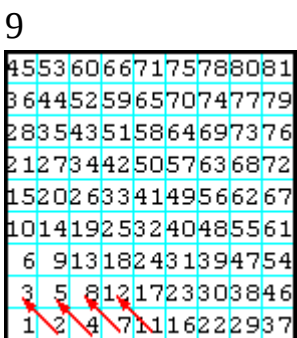
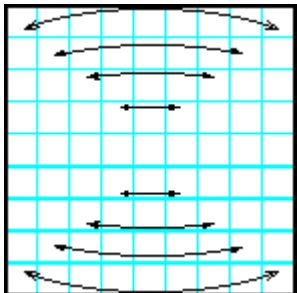
Звільнення пам'яті аналогічне як при роботі з функцією `malloc()`.

Контрольні запитання:

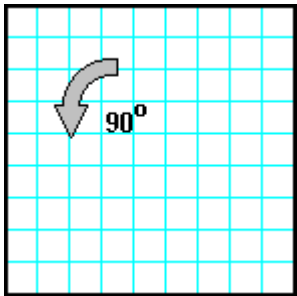
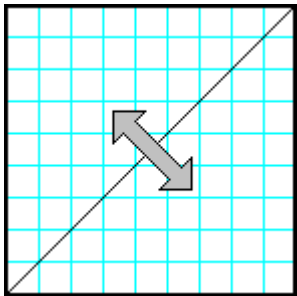
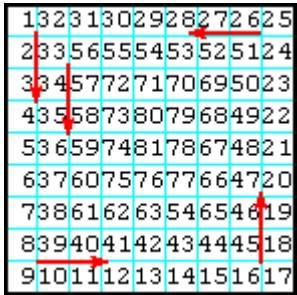
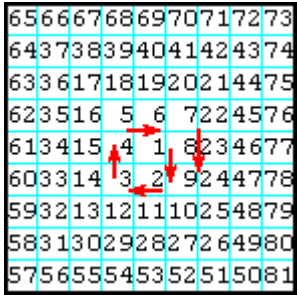
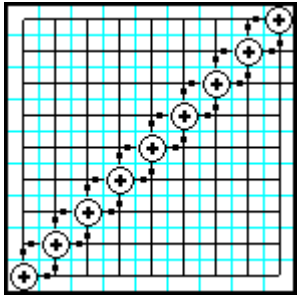
- 1) Для чого використовуються динамічні масиви?
- 2) З допомогою яких функцій можна створити динамічний масив?
- 3) З допомогою якої функції звільняється пам'ять під виділений масив?
- 4) Яким чином можна звертатись до елементів динамічного масиву?
- 5) Що таке адресний вираз при звертання до елементів масиву?

Індивідуальні завдання:

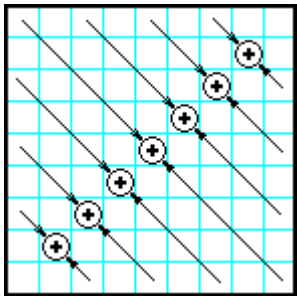
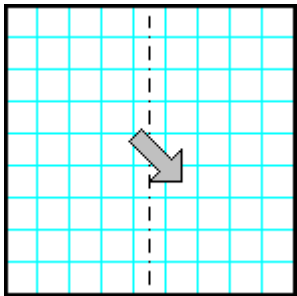
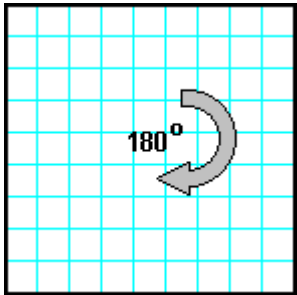
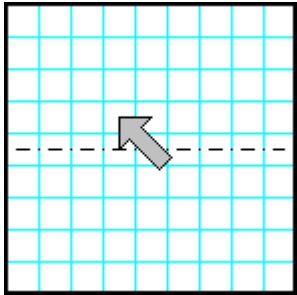
№	Завдання	Ілюстрація
1	Заповнити матрицю випадковими числами. Повернути матрицю на 90° за годинниковою стрілкою.	
2	Заповнити матрицю випадковими числами. Відобразити матрицю симетрично відносно головної діагоналі.	
3	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від лівого верхнього кута по спіралі: вправо – вниз – вліво – вгору. (Примітка. На малюнку вказані індекси елементів вектора b)	
4	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від центра по спіралі: вліво – вниз – вправо – вгору. (Примітка. На малюнку вказані індекси елементів вектора b)	
5	Заповнити матрицю випадковими числами. На головній діагоналі розмістити суми елементів, які лежать у тому ж рядку і у тому ж стовпці.	

№	Завдання	Ілюстрація
6	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від лівого верхнього кута по діагоналі: вправо - вгору. (Примітка. На малюнку вказані індекси елементів вектора b)	
7	Заповнити сектори матриці, які лежать вліво і вправо від головної та бічної діагоналей, значеннями вектора $b_1, b_2, \dots, b_{32}$, від лівого верхнього кута вниз – вправо. Решта елементів матриці заповнити нулями. (Примітка. На малюнку вказані індекси елементів вектора b)	
8	Заповнити матрицю випадковими числами. Відобразити матрицю симетрично відносно вертикальної осі сектори матриці, які лежать вліво і вправо від головної і бічної діагоналей.	
9	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від лівого нижнього кута по діагоналі: вліво – вгору. (Примітка. На малюнку вказані індекси елементів вектора b)	
10	Заповнити матрицю випадковими числами. Відобразити головну і бічну діагоналі симетрично відносно вертикальної осі.	

№	Завдання	Ілюстрація
11	Заповнити матрицю випадковими числами. Розмістити на головній діагоналі суми елементів, котрі лежать на діагоналях, перпендикулярних до головної.	
12	Заповнити матрицю випадковими числами. Відобразити верхню половину матриці на нижню дзеркально симетрично відносно горизонтальної осі.	
13	Заповнити матрицю випадковими числами. Розбити матрицю на квадрати розміром 3x3. В центрі кожного квадрата помістити суму решти елементів квадрата.	
14	Заповнити матрицю випадковими числами. Відобразити праву половину матриці на ліву дзеркально симетрично відносно вертикальної осі.	
15	Заповнити сектори матриці, які лежать вліво і вправо від головної та бічної діагоналей значеннями вектора $b_1, b_2, \dots, b_{32}$ від лівого верхнього кута вправо – вниз. Решту матриці заповнити нулями. (Примітка. На малюнку вказані індекси елементів вектора b)	

№	Завдання	Ілюстрація
16	Заповнити матрицю випадковими числами. Повернути матрицю на 90° проти годинникової стрілки.	
17	Заповнити матрицю випадковими числами. Відобразити матрицю симетрично відносно бічної діагоналі.	
18	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від лівого верхнього кута по спіралі: вниз – вправо – вверх – вліво. (Примітка. На малюнку вказані індекси елементів вектора b)	
19	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від центра по спіралі: вниз – вліво – вверх – вправо. (Примітка. На малюнку вказані індекси елементів вектора b)	
20	Заповнити матрицю випадковими числами. На бічній діагоналі розмістити суми елементів, які лежать у тому ж рядку і у тому ж стовпці.	

№	Завдання	Ілюстрація
21	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від лівого верхнього кута по діагоналі: вліво – вниз. (Примітка. На малюнку вказані індекси елементів вектора b .)	
22	Заповнити сектори матриці, які лежать вище і нижче від головної та бічної діагоналей, значеннями вектора $b_1, b_2, \dots, b_{32}$, від лівого верхнього кута вправо – вниз. Решта матриці заповнити нулями. (Примітка. На малюнку вказані індекси елементів вектора b .)	
23	Заповнити матрицю випадковими числами. Відобразити матрицю симетрично відносно горизонтальної осі сектори матриці, які лежать вище і нижче від головної і бічної діагоналей.	
24	Заповнити матрицю значеннями вектора $b_1, b_2, \dots, b_{81}$ від правого верхнього кута по діагоналі: вліво – вниз. (Примітка. На малюнку вказані індекси елементів вектора b .)	
25	Заповнити матрицю випадковими числами. Відобразити головну і бічну діагоналі симетрично відносно горизонтальної осі.	

№	Завдання	Ілюстрація
26	Заповнити матрицю випадковими числами. Розмістити на бічній діагоналі суми елементів, котрі лежать на діагоналях, перпендикулярних до бічної.	
27	Заповнити матрицю випадковими числами. Відобразити ліву половину матриці на праву дзеркально симетрично відносно вертикальної осі.	
28	Заповнити матрицю випадковими числами. Повернути матрицю на 180° за годинниковою стрілкою.	
29	Заповнити матрицю випадковими числами. Відобразити нижню половину матриці на верхню дзеркально симетрично відносно горизонтальної осі.	
30	Заповнити сектори матриці, які лежать вище і нижче від головної та бічної діагоналей, значеннями вектора $b_1, b_2, \dots, b_{32}$, від лівого верхнього кута вниз – вправо. Решта матриці заповнити нулями. (Примітка. На малюнку вказані індекси елементів вектора b)	