

Лабораторна робота №8

Робота з динамічними масивами та функціями

1. Ціль роботи:

- 1) Отримання практичних навичок при роботі з масивами та покажчиками.
- 2) Отримати практичні навички виділення, перерозподілу і звільнення пам'яті при роботі з динамічними масивами

2. Завдання

Завдання 8.1. Нехай є наступний фрагмент програми, поданих у варіантах (табл. 8.1). Поясніть як зміниться масив після його виконання, при наступному початку програми (Вказівка: замість N підставити номер варіанта за списком групи).

```
#include <iostream>
#include<cmath>
using namespace std;
int* form(int&n) {  n=10+N%10;
    int*a=new int[n];
    a[0]=N;
    for(int i=1;i<n;i++)
        a[i]=a[i-1]+1;
    return a;
}
int main() {
```

Таблиця 2.1 – Варіанти завдання 2.1

№	Фрагмент програми	№	Фрагмент програми
1-5	<pre>int *a, n; a=form(n); int **p=new int *[n]; for (int i=0;i<n;i++) { p[i]=(a+i); *p[i]=*p[i]+N; }</pre>	6-10	<pre>int *a, n,i,*p; a=form(n); for (p = &a[0], i = 0 ; i <n; i++) { p[i] -=N*N; }</pre>
11-15	<pre>int *a, n,i,*p; a=form(n); for (p = a, i = 0; p+i < a+n; i++) { p[i] *=*(p+i); }</pre>	16-20	<pre>int *a, n,i,*p; a=form(n); for (p = a+n, i=1; i <=n; i++) { *(p-i) *= (int)sqrt(N); }</pre>
21-25	<pre>int *a, n,i,*aa; a=form(n); int **p=new int *[n]; aa=a+n-1; for (i=0;aa-a>=0;aa--,i++) { p[i]=aa; *p[i]+=a[i]; }</pre>	26-30	<pre>int *a, n,i,t=0,*p; a=form(n); int *aa=new int[n]; p=aa; for (i=n-1;i>=0;i--) { *p++=a[i]; t+=a[i]; } t=t/n; for (i=0;i<n;i++) a[i]=*(aa+i)+t;</pre>

Завдання 8.2. Скласти програму, що заповнює динамічний одновимірний масив за допомогою випадкових чисел, потім змінює масив, згідно завдань, які наведені в таблиці 8.2.

Завдання 8.3. Скласти програму, що заповнює динамічний двовимірний масив за допомогою випадкових чисел, потім змінює масив, згідно завдань, які наведені в таблиці 8.2.

Таблиця 8.2

№ вар.	Одновимірний масив	Двовимірний масив
1	Додати K простих чисел на початку масиву	Додати рядок із заданим номером
2	Видалити перший двозначний від'ємний елемент	Додати стовпець із заданим номером
3	Видалити елемент із заданим ключем (значенням)	Додати рядок в кінець матриці
4	Видалити елемент, який дорівнює середньому арифметичному елементів масиву	Додати стовпець в кінець матриці
5	Видалити елемент із заданим номером	Додати рядок на початку матриці
6	Видалити N елементів, починаючи з номера K	Додати стовпець на початку матриці
7	Видалити всі парні елементи	Додати K рядків в кінець матриці
8	Видалити всі елементи з парними індексами	Додати K стовпців в кінець матриці
9	Видалити всі непарні елементи	Додати K рядків на початку матриці
10	Видалити всі елементи з непарними індексами	Додати K стовпців на початку матриці
11	Додати K чисел Фібоначчі на початок масиву	Видалити рядок з номером K
12	Додати K чисел Фібоначчі в кінець масиву	Видалити стовпець з номером K
13	Додати K елементів в початок масива	Видалити рядки, починаючи з рядка K1 і до рядка K2
14	Додати K елементів в кінець масива	Видалити стовпці, починаючи зі стовпця K1 і до стовпчика K2
15	Додати K елементів, починаючи з номера N	Видалити всі парні рядки
16	Додати після кожного від'ємного елемента його модуль	Видалити всі парні стовпці
17	Додати після кожного парного елемента елемент зі значенням 0	Видалити всі рядки, в яких є хоча б один нульовий елемент
18	Додати по K елементів в початок і в кінець масиву	Видалити всі стовпці, в яких є хоча б один нульовий елемент
19	Додати K простих чисел в кінець масиву	Видалити рядок, в якій знаходиться найбільший елемент матриці
20	Видалити елемент із заданим номером	Додати рядки після кожного парного рядка матриці
21	Видалити N елементів, починаючи з номера K	Додати стовпці після кожного парного стовпця матриці
22	Видалити всі парні елементи	Додати K рядків, починаючи з рядка з номером N

23	Видалити всі елементи з парними індексами	Додати K стовпців, починаючи зі стовпчика з номером N
24	Видалити всі непарні елементи	Додати рядок після рядка, що містить найбільший елемент
25	Видалити всі елементи з непарними індексами	Додати стовпець після стовпчика, що містить найбільший елемент

3. Методичні вказівки

1. Для виділення пам'яті під масиви використовувати операцію new, для видалення масивів з пам'яті - операцію delete.
2. Для виділення пам'яті, заповнення масивів, видалення і додавання елементів (рядків, стовпців) написати окремі функції. У функції main () повинні бути розміщені тільки опис змінних і звернення до відповідних функцій:

```
int main()
{
    int n;
    int *mas=formMas(n);
    initMas(mas,n);
    printMas(mas,n);

    .....
    return 0;
}
```

3. При видаленні елементів (рядків, стовпців) передбачити помилкові ситуації, т. б. ситуації, в яких буде виконуватися спроба видалення неіснуючого елемента (рядка, стовпця) або кількість елементів, що видаляються, буде перевищувати кількість наявних елементів (рядків, стовпців). У цьому випадку має бути виведено повідомлення про помилку.