

Лабораторна робота *з дисципліни «Алгоритмізація та програмування» № 6*

Тема: Програмне опрацювання одновимірних масивів

Мета роботи: Вивчити засоби мови C++ для оголошення та ініціалізації статичних одновимірних масивів, звернення до їх елементів, введення масивів з клавіатури, генерування випадкових значень елементів та виведення їх на екран.

Теоретичні відомості

Оголошення

Одновимірний масив – це скінчена іменована область пам'яті, в якій зберігається група однотипних величин, що розрізняються за допомогою індексу.

Оголошення масиву подібне до оголошення звичайної змінної, лише для оголошення звичайного одновимірного масиву слід вказати кількість елементів в квадратних дужках після імені масиву

Одновимірний масив (вектор) оголошується так:

`<тип_даних> <ім'я_масива> [<розмір_масива>];`

де тип даних – це тип елементів масива (елементами масива не можуть бути функції та елементи типу void); розмір масива – визначає кількість елементів масива.

Значення розміру масива при його оголошенні не вказують у таких випадках:

- коли при оголошенні масив ініціалізовується;
- коли масив оголошено як формальний параметр функції;
- коли масив оголошено як посилання на масив, явно визначений в іншому модулі.

Наприклад, оголошення одновимірного масиву з 10 елементів цілого типу може мати вигляд

```
int a[10];
```

Елементи масиву нумеруються, починаючи від нуля, тобто масив а складається із 10-ти елементів цілого типу, індекс (номер) яких мусить мати значення від 0 до 9.

Кількість елементів масиву разом з типом його елементів визначає об'єм пам'яті, необхідний для розміщення масиву, – цей об'єм визначається на етапі компіляції, і тому кількість елементів масиву може бути вказана лише цілою додатною константою або константним виразом.

Оголошення деяких масивів:

```
int b[5];      // 5 цілих чисел: a[0], a[1],..., a[4]
float m[30];   // 30 дійсних чисел: m[0], m[1],..., m[29]
double v[15];  // 15 дійсних чисел з подвійною точністю: v[0], v[1],..., v[14]
char N[20];    // 20 символів: N[0], N[1],..., N[19]
```

Ініціалізація

При оголошенні масивів їхнім елементам можна присвоювати початкові значення. Якщо реальна кількість ініціалізованих значень є меншою за розмір масива, то решта елементів масива отримують значення 0:

```
int a[5] = { 9, 33, 23, 8, 1 }; // a[0]=9, a[1]=33, a[2]=23, a[3]=8, a[4]=1
int a[] = { 9, 33, 23, 8, 1 }; // аналогічно, довжина масиву визначається компілятором
float b[10] = { 1.5, -3.8, 10 }; // b[0]=1.5, b[1]=-3.8, b[2]=10, b[3]= ... =b[9]=0
```

Всі дії над масивами виконуються по-елементно, мова C / C++ не дозволяє присвоєння масивів один одному.

Доступ до елементів

Для доступу до елемента слід вказати значення індексу, записане в квадратних дужках після імені масиву:

`<ім'я_масива> [<значення_індекса>]`

Наприклад: `a[i]`. Елементи масиву нумеруються, починаючи від нуля; останній елемент масиву має індекс, на одиницю менший вказаної при оголошенні кількості елементів.

Для доступу до елементів одновимірного масива за допомогою вказівника використовується така форма:

`*(<ім'я_масива> + <індекс>)`

Під час програмної реалізації індексний спосіб доступу до елемента масиву (`<ім'я_масива>[<індекс>]`) зводиться до адресного (вказівникового), тобто індексний вираз перетворюється в адресний, а оскільки операції над вказівниками виконуються швидше і, якщо елементи масиву опрацьовуються по чергово, то доцільніше використовувати другий спосіб. Якщо ж вибір елементів є випадковим, то, щоб уникнути помилок, краще використовувати перший спосіб. Крім того, перший спосіб є більш наочним і зручнішим для сприйняття, що сприяє кращому читанню програм.

Оскільки ім'я масива є вказівником на перший елемент масива, то вказівнику можна присвоїти адресу першого елемента масива за допомогою такого оператора присвоєння:

```
int* ptr;
ptr = m;
```

Цей вираз аналогічний присвоєнню адреси першого елемента масиву, тобто елемента з нульовим індексом: `ptr = &m[0];`

Також для оголошення масивів можна використовувати типізовані констант-масиви, які дають змогу водночас і оголосити масив, і задати його значення як константи:

```
const <тип_даних> <ім'я_масива> [<розмір_масива>] = {<значення_елементів_масива>;};
```

Введення одновимірних масивів

Введення масива вручну:

```
const int colCount = 5;
for (int i = 0; i < colCount; i++)
{
    cout << "a[" << i << "]="; cin >> a[i];
}
```

При цьому числа, що вводяться можна записувати як у стовпчик, так і в рядок через пробіл.

Введення масива за допомогою генератора випадкових чисел:

(кожного разу генерується інша послідовність чисел):

```
const int colCount = 5, Low = 6, High = 20;
int a[colCount];
srand((unsigned)time(NULL));
for (int i = 0; i < colCount; i++)
    a[i] = Low + rand() % (High - Low + 1);
```

Виведення одновимірних масивів

У стовпчик:

```
for (int i = 0; i < colCount; i++)
    cout << "a[" << i << "]= " << a[i] << endl;
```

У рядок

```
for (int i = 0; i < colCount; i++)
    cout << "a[" << i << "]= " << a[i] << "\t" << endl;
// або розділяти значення за допомогою форматування
// cout << setw(5) << "a[" << i << "][" << j << "]= " << setw(2) << a[i][j];
```

Передавання масивів у функцію

При передаванні масиву у функцію передається адреса його першого елемента. Таким чином, масив завжди передається за адресою, а не за значенням. При цьому інформація про кількість елементів масиву – втрачається, і тому кількість елементів слід передавати за допомогою окремого параметру.

Масив можна передати у функцію за допомогою формального параметра різними способами. При використанні масива як параметра у функцію передається вказівник на його перший елемент, тобто масив завжди передається за адресою. При цьому інформація про кількість елементів масива втрачається, тому його розмірність потрібно передавати через окремий параметр.

Нижче наведений приклад функції, що приймає масив цілих чисел та повертає суму його елементів.

```
#include <iostream>
using namespace std;

int Sum(int arr[], const int n); // прототип функції

int main()
{
    const int k = 10; // кількість елементів масиву
    int a[k] = { 3,4,5,7,6 }; // оголосили та ініціалізували масив
    cout << "S = " << Sum(a, k) << endl; // виклик функції
    return 0;
}

int Sum(int arr[], const int n) // функція для обчислення суми елементів
{
    int s = 0;
    for (int i = 0; i < n; i++)
        s += arr[i];
    return s;
}
```

У наступному прикладі реалізоване аналогічне завдання, але масив заповнений випадковими числами і виведений на екран перед викликом функції.

```
#include <iostream>
using namespace std;

int sumPointer(const int* arr, int Columns)
{
    int s = 0;
    for (int i = 0; i < Columns; i++)
        s += arr[i]; // обчислення суми
    return s;
}

void main() {
    // розмір масива
    const int colCount = 5;
```

```

int b[colCount];
const int Low = 6, High = 20; // для заповнення масиву випадковим чином
srand((unsigned)time(NULL));
//заповнення елементів значеннями
for (int i = 0; i < colCount; i++)
    b[i] = Low + rand() % (High-Low+1);
// вивід отриманого масиву
for (int i = 0; i < colCount; i++)
    cout << b[i]<< endl;
// виклик функції
cout << "Array's elements Sum: " << sumPointer(b, colCount) << endl;
}

```

Повернення масивів з функцій

Щоб функція повертала одновимірний масив, потрібно її оголосити як функцію, що повертає вказівник:

```

int * myFunction()
{ ...
}

```

Наприклад, розглянемо функції, що ініціалізує одновимірний масив випадковими числами і повертає його як результат.

```

#include <iostream>
using namespace std;

int* init(int* arr, int Columns) {
    const int Low = 6, High = 20;
    srand((unsigned)time(NULL));
    for (int i = 0; i < Columns; i++)
        arr[i] = Low + rand() % (High - Low + 1);
    return arr;
}

void main() {
    // розмір масива повинен бути константою
    const int colCount = 5;
    int b[colCount];
    int* p = init(b, colCount); // виклик функції

    // вивід отриманого масиву
    for (int i = 0; i < colCount; i++)
        cout << p[i]<< endl;
}

```

Завдання 1

1. Написати програму для виконання певних дій над статичним одновимірним масивом, реалізувавши доступ до елементів масива двома способами: за допомогою індексів та вказівників. Кожний спосіб реалізувати окремою функцією.
2. Алгоритм формування початкового масива реалізувати двома способами: за допомогою введення даних з клавіатури та випадкової генерації чисел. Кожний спосіб реалізувати окремою функцією.
3. Вивести на екран елементи початкового масива. Якщо є кілька початкових масивів, то вивести усі. Якщо потрібно визначити певні індекси, значення тощо, то вивести їх. Якщо масив був змінений, то вивести модифікований масив.
4. Використання глобальних змінних у підпрограмах не допускається. Інформація у підпрограми повинна передаватися лише за допомогою параметрів.

5. Введення даних, виведення даних і виконання певних дій над масивами потрібно реалізувати в окремих функціях. У головній програмі потрібно виконувати лише їхній виклик. Введення-виведення даних супроводжувати відповідними повідомленнями.
6. Побудувати блок-схему алгоритму програми та блок-схеми алгоритмів усіх функцій з індексним доступом до елементів масива.

№ з/п	Завдання
1.	Написати підпрограму, яка обчислює середнє арифметичне непарних елементів одновимірного масиву (вектора) із n елементів цілого типу.
2.	Написати підпрограму, яка міняє місцями перший елемент із найменшим парним елементом одновимірного масиву (вектора) із n елементів цілого типу.
3.	Написати підпрограму, яка шукає максимальний та мінімальний елементи одновимірного масиву (вектора) із n елементів цілого типу.
4.	Написати підпрограму, яка міняє місцями елементи одновимірного масиву (вектора) із n елементів цілого типу так, щоб вони розмістилися в зворотному порядку: $a_n, a_{n-1}, \dots, a_2, a_1$.
5.	Написати підпрограму, яка обчислює суму індексів непарних елементів одновимірного масиву (вектора) із n елементів цілого типу.
6.	Написати підпрограму, яка обчислює середнє арифметичне елементів одновимірного масиву (вектора) із n елементів цілого типу з парними індексами.
7.	Написати підпрограму, яка міняє місцями останній елемент із найбільшим непарним елементом одновимірного масиву (вектора) із n елементів цілого типу.
8.	Написати підпрограму, яка шукає індекси найбільшого та найменшого елементів одновимірного масиву (вектора) із n елементів цілого типу.
9.	Написати підпрограму, яка обчислює середнє арифметичне максимального та мінімального елементів одновимірного масиву (вектора) із n елементів цілого типу.
10.	Написати підпрограму, яка обчислює суму елементів одновимірного масиву (вектора) із n елементів цілого типу з непарними індексами.
11.	Написати підпрограму, яка обчислює суму максимального та мінімального елементів одновимірного масиву (вектора) із n елементів цілого типу.
12.	Написати підпрограму, яка шукає найменший парний елемент одновимірного масиву (вектора) із n елементів цілого типу.

13.	Написати підпрограму, яка шукає індекс найбільшого парного елемента одновимірного масиву (вектора) із n елементів цілого типу.
14.	Написати підпрограму, яка обчислює суму індексів максимального та мінімального елементів одновимірного масиву (вектора) із n елементів цілого типу.
15.	Написати підпрограму, яка шукає найбільший непарний елемент одновимірного масиву (вектора) із n елементів цілого типу.
16.	Написати підпрограму, яка обчислює середнє арифметичне індексів парних елементів одновимірного масиву (вектора) із n елементів цілого типу.
17.	Обчислити середнє арифметичне максимального та мінімального елементів одновимірного масиву (вектора) із n елементів цілого типу.
18.	Замінити від'ємні елементи масиву, розташовані у непарних стовпчиках, їхніми модулями та знайти суму цих елементів.
19.	Обчислити добуток елементів масиву, розташованих після максимального за модулем елемента.
20.	Обчислити суму елементів масиву, розташованих після першого додатного елемента.

Завдання 2

- Написати програму для виконання певних дій над статичним одновимірним масивом, реалізувавши доступ до елементів масива двома способами: за допомогою індексів та вказівників. Кожний спосіб реалізувати окремою функцією.
- Алгоритм формування початкового масива реалізувати двома способами: за допомогою введення даних з клавіатури та випадкової генерації чисел. Кожний спосіб реалізувати окремою функцією.
- Вивести на екран елементи початкового масива. Якщо є кілька початкових масивів, то вивести усі. Якщо потрібно визначити певні індекси, значення тощо, то вивести їх. Якщо масив був змінений, то вивести модифікований масив.
- Використання глобальних змінних у підпрограмах не допускається. Інформація у підпрограми повинна передаватися лише за допомогою параметрів.
- Введення даних, виведення даних і виконання певних дій над масивами потрібно реалізувати в окремих функціях. У головній програмі потрібно виконувати лише їхній виклик. Введення-виведення даних супроводжувати відповідними повідомленнями.
- Побудувати блок-схему алгоритму програми та функції, що виконує вказані дії над масивом.

№ з/п	Завдання
----------	----------

1.	Визначити значення та порядкові номери двох найбільших елементів вектора дійсних чисел.
2.	Визначити значення та порядкові номери максимального від'ємного елемента вектора дійсних чисел.
3.	Визначити значення та порядкові номери мінімального додатного елемента вектора цілих чисел.
4.	Включити у відсортований за спаданням значень масив цілих чисел введене з клавіатури ціле число так, щоб не порушити загальну впорядкованість елементів масива.
5.	Серед елементів першого вектора дійсних чисел, які не входять у другий вектор дійсних чисел, знайти значення та порядкові номери максимального елемента.
6.	Відсортувати елементи сформованого із двох векторів цілих чисел вектора за зростанням.
7.	Записати у два масиви усі дільники двох цілих чисел. Крім того, спільні дільники записати у третій масив. Утворити число, дільниками якого є числа, записані у третій масив, і вивести його на екран.
8.	Цифри цілої частини дійсного числа записати у масив. Утворити та вивести на екран ціле число зі зворотним порядком цифр.
9.	Вивести на екран елементи одновимірного масива цілих чисел, значення яких зустрічається більше одного разу.
10.	Визначити кількість повторень кожного елемента одновимірного масива

	цілих чисел. Елементи масива можуть повторюватися.
11.	Ввести масив цілих додатних десяткових чисел. Виконати переведення кожного числа з десяткової системи числення у двійкову. Для переведення використати алгоритм виділення залишків від цілочисельного ділення десяткового числа на 2. Двійкові зображення чисел вивести на екран.
12.	Циклічно зсунути всі елементи масива цілих чисел на K позицій вліво. Значення K ввести з клавіатури. При циклічному зсуві вліво початковий елемент масива записується на місце зсунутого кінцевого елемента.
13.	Створити новий масив, вибравши з кожної пари двох сусідніх елементів одновимірного масива цілих чисел максимальний. Визначити значення мінімального елемента створеного масива.
14.	Перевірити, чи вектор цілих чисел є спадаючим. Якщо так, то записати елементи масива у порядку зростання (без застосування алгоритму сортування), інакше – відсортувати його за спаданням.

15.	Перші 10 простих чисел натурального ряду записати в одновимірний масив. Визначити середнє арифметичне значення елементів цього масива.
16.	Сформувати вектор з 20 непарних чотирицифрових випадкових чисел без повторень елементів. Визначити мінімальний і максимальний елементи вектора та поміняти їх місцями.
17.	Сформувати вектор з усіх простих чисел, що потрапляють у заданий інтервал (a,b). Межі інтервала ввести з клавіатури. Сформований масив вивести на екран.
18.	Ввести масив цілих чисел та циклічно зсунути усі його елементи на K позицій вправо. Значення K утворюється випадковим чином і є числом, що не перевищує 10. При циклічному зсуві вправо кінцевий елемент масива записується на місце зсунутого початкового елемента.
19.	Сформувати вектор із N цілих випадкових чисел, які приймають значення на відрізку [a,b]. Межі відрізка ввести з клавіатури. Визначити середнє арифметичне значення елементів масива. Знайти квадратичне відхилення кожного числа від середнього значення та записати його у новий вектор. Знайти значення та номери елементів з найбільшим відхиленням.
20.	Сформувати масив з 10 випадкових цілих чисел, що належать заданому інтервалу (a,b]. Межі інтервала ввести з клавіатури. Елементи масива не повинні повторюватися. Відсортувати сформований масив за спаданням значень елементів.
30.	Створити одновимірний масив, записавши у нього менші елементи з кожної пари елементів двох масивів дійсних чисел однакової розмірності. Кожна пара елементів визначається однаковим значенням індексів. Знайти максимальний елемент утвореного масиву.