

Лабораторна робота № 7.

Тема заняття: Одновимірні масиви.

Мета: Одержання практичних навичок створення та використання одновимірних масивів в C++.

Зміст роботи.

Написати програми мовою C++:

Завдання 1

1. Задати масив із 5 елементів. Масив заповнити з клавіатури. Вивести масив на екран у зворотньому порядку.
2. Дано цілочисловий одновимірний масив, який складається з 16 елементів. Знайти парні елементи масиву, які більші числа А, і їх кількість. Число А ввести з клавіатури.

Завдання 2

Скласти програму і розв'язати задачу згідно варіанту. Вхідні дані підібрати самостійно з метою перевірки правильності розробленого алгоритму. Дано цілочисловий одновимірний масив, який складається з 10 елементів.

1. Обчислити суму додатних та від'ємних елементів масиву, підрахувати їх кількість та визначити яких елементів в масиві більше.
2. Обчислити добуток додатних, кратних числу 3 елементів та їх кількість. Визначити скільки елементів масиву не відповідають даній умові.
3. Обчислити квадратний корінь суми квадратів елементів та визначити суму квадратів додатних елементів.
4. Обчислити суму попарних добутків елементів (скалярний добуток) двох масивів.
5. Обчислити середньоарифметичне від'ємних елементів. Від'ємні елементи вивести на екран.
6. Визначити кількість елементів, більших ніж середньоарифметичне всіх елементів.
7. Визначити кількість елементів, більших ніж деяке задане число, яке треба ввести з клавіатури.
8. Задати числовий інтервал і визначити суму значень елементів, які потрапляють у цей інтервал.
9. Додатні елементи зменшити на 5, а від'ємні збільшити на число 2. Обчислити суму елементів. Визначити, яких елементів в масиві більше.
10. Визначити скільки в масиві є від'ємних, додатних та нульових елементів. Підрахувати середнє значення елементів масиву.
11. Визначити номер найменшого елемента серед додатних елементів масиву.
12. Визначити суму елементів з парними індексами і добуток з непарними. Добутки порівняти.
13. Утворити масив з елементами $d_i = 2 * a_i$. Обчислити добуток додатних елементів та підрахувати їх кількість.

14. Визначити середньоарифметичне значення елементів що знаходяться в заданому користувачем проміжку.
15. У якому з двох масивів більше додатних елементів?

Методичні рекомендації.

Масив – це складений тип даних, що складається з фіксованого числа елементів одного й того ж типу даних, який називають базовим.

Характерні риси масивів:

- тип елементів масива повинен бути конкретно описаний;
- масив – це структура з так званим прямим доступом: всі її компоненти можуть вибиратися довільно й однаково доступні;
- для позначення окремого елемента масива необхідно знати його порядковий номер у структурі – **індекс елемента масива**;
- число елементів масива задається при його описі й надалі не змінюється.

Синтаксис оголошення одновимірного масива:

Тип_елементів ім'я_масива [кількість_елементів];

Індекс елементів масива починається з 0 (нуль); останній індекс дорівнює K-1, тобто число елементів масива на 1 більше від індексу останнього елемента масива.

У мові програмування C++ всі масиви займають суміжні елементи пам'яті. Іншими словами, елементи масива в пам'яті розташовані послідовно один за одним. Клітина з найменшою адресою належить до першого елемента масива, а з найбільшою – до останнього.

Приклад . Ініціалізація одновимірного масиву квадратами чисел від 1 до 10.

```
int main()
{
    int A[10];
    setlocale(LC_ALL,"Ukrainian");
    for(int t=0; t<10; ++t)
        A[t] = (t+1)*(t+1);
    // Відображається масив.
    for(int t=0; t<10; ++t)
        cout << A[t] <<" ";
    return 0;
}
```

Генерація випадкових чисел.

Для генерації псевдовипадкових чисел можна використовувати функцію rand, яка повертає значення цілого типу в діапазоні від 0 до максимального значення типу int. Кожне звертання до функції у процесі виконання програми викликає генерування нового псевдовипадкового числа з рівномірним законом розподілу.

Загальний діапазон чисел, які можна отримати викликаючи функцію rand: 0 – 32767. Якщо дуже добре подумати, то ваші знання дозволяють керувати

діапазонами. Тут нам знадобиться ділення по модулю. Візьмемо будь-яке число, для прикладу 57. І число x . Якщо $x \% 52 = 0$ (тобто $x = 52$), це мінімальне значення, яке ми можемо досягти. Всі інші будуть мати певний залишок і при цьому цей залишок не може бути більшим 51. Ось і все.

```
rand()%52; // Діапазон від 0 до 51
```

Добре, ми навчилися встановлювати діапазони з 0 до . Але радіти ще рано, нам може знадобитися діапазон від x до $(x+15)$ (для прикладу, де $x > 0$).

Допустимо нам потрібен діапазон від 15 до 24. Для цього потрібно зробити декілька кроків:

1. $24 - 15 = 9$
2. Згенерувати числа від 0 до 9. Тобто, `rand() % 10;`
3. Тепер посунути це все на 15 одиниць. `rand() % 10 + 15;`

Приклад використання функції `rand`:

```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
    srand(unsigned(time(NULL)));
    for ( int i = 0; i < 10; ++i )
        cout << rand() << endl;
}
```

Функція `rand` починає генерування значень з єдиного встановленого початкового значення. Тому при виконання програми кожного разу будемо отримувати одну і ту саму послідовність псевдовипадкових чисел. Для того, щоб послідовності чисел були різними, необхідно встановлювати їх початкове значення. Для цього можна використовувати функцію `time`, яка повертає поточний час.

Для генерації випадкових чисел в довільному діапазоні значень використовуються операцію `%`. Наприклад, необхідно отримати 10 чисел в діапазоні від 5 до 15:

```
#include <iostream>
#include <ctime>
using namespace std;
int main() {
    srand(unsigned(time(NULL)));
    for ( int i = 0; i < 10; ++i )
        cout << 5 + rand() % 10 << endl; }
```

```
randomDigits[i] = rand() % 7;          // 0 ... 6
randomDigits_2[i] = 1 + rand() % 7;    // 1 ... 7
randomDigits_3[i] = 200 + rand() % 101; // 200 ... 300
randomDigits_4[i] = rand() % 41 - 20;  // -20 ... 20
randomDigits_5[i] = 0.01 * (rand() % 101); // 0.01 ... 1
}
```

Контрольні запитання.

1. Що таке масив ?
2. Як оголосити масив?
3. Поясніть принцип розміщення масивів у пам'яті.
4. Якими способами можна ініціювати масив?
5. Як здійснюється звернення до елементів масивів?
6. Як у C++ реалізується введення-виведення елементів масиву?
7. Як організувати перегляд масиву
8. Як змінити значення елементів масиву?