Лабораторна робота №4. Робота з масивами та строками

Мета: засвоїти методи роботи з масивами та строками в С++.

Теоретичні відомості

Поняття масиву

Масив – це структура даних, яка містить скінченний набір даних одного типу. Доступ до елементів масивів виконується за допомогою індексів. Можливе використання масивів будь-якого типу даних, в тому числі і масиви масивів. Наприклад, строки – це масиви символів.

Кожне значення зберігається в окремому елементі масива. В оперативній пам'яті всі елементи зберігаються послідовно.

Для оголошення масиву потрібно вказати:

- тип значення кожного елементу;
- ім'я масиву;
- кількість елементів масиву.

Синтаксис оголошення масиву наступний:

```
ім'я типу ім'я масива [розмір масива];
```

Важливою особливістю використання масивів в мові С++ є те, що перший елемент масива має індекс 0.

Приклад 1. Ініціалізація масива та дії над його елементами.

```
//example 4_1.cpp Робота з масивами
//Приклад 1 до лабораторної роботи №4
//Автор: Кудін О.В., 01.10.2017
#include "stdafx.h"
#include <iostream>
int main()
{
	using namespace std;
	int yams[3];	// оголошення масива з трьома елементами
	yams[0] = 7;	// ініціалізація елементів масива
	yams[1] = 8;
	yams[2] = 6;

	int yamcosts[3] = { 20, 30, 5 }; // оголошення та ініціалізація масива

	cout << "Total yams = ";
	cout << yams[0] + yams[1] + yams[2] << endl;
	cout << "The package with " << yams[1] << " yams costs ";
	cout << yamcosts[1] << " cents per yam.\n";
	int total = yams[0] * yamcosts[0] + yams[1] * yamcosts[1];
	total = total + yams[2] * yamcosts[2];
	cout << "The total yam expense is " << total << " cents.\n";
	cout << "\nSize of yams array = " << sizeof yams;
	cout << " bytes.\n";
	cout << "Size of one element = " << sizeof yams[0];
	cout << " bytes.\n";
	return 0;
}
```

Існує декілька правил ініціалізації масивів, яких слід дотримуватись при написанні програм.

Ініціалізація виконується **тільки** під час оголошення.

```
int cards[4]={1,2,3,4};          // правильно
int hand[4];                     // правильно
hand[4]={5,6,7,8};               // помилка
hand=cards;                      // помилка
```

При ініціалізації масива можливо вказувати менше значень, ніж вказано при оголошенні.

```
int nums[10]={1,2,3,4};
```

При частковій ініціалізації всім іншим елементам буде присвоєно 0.

Можливо не вказувати кількість елементів масива при оголошенні. Тоді компілятор сам порахує кількість елементів, яка вказана при ініціалізації, при цьому, кількість елементів для програми, краще розрахувати та записати в окрему змінну. Наприклад, це можна зробити так:

```
int arr[]={1,2,3,4,5,23,34};
int len_arr=sizeof arr/ sizeof (int);
```

Багатовимірні масиви

Багатовимірні масиви задаються вказанням при оголошенні масива кожного виміру у квадратних дужках.

Синтаксис оголошення масиву наступний:

```
ім’я типу ім’я масива[розмір виміру1][розмір виміру2]…[розмір вимірuN];
```

Наприклад, оголошення двовимірного масиву має вигляд

```
int matr[6][8];
```

Тут задається двовимірний масив з 6 строк та 8 стовпців.

Приклад 2. Визначення номер строки масива, яка містить максимальну кількість нульових елементів.

```
//example 4_2.cpp Робота з двовимірним масивом
//Приклад 2 до лабораторної роботи №4
//Автор: Кудін О.В., 01.10.2017
#include "stdafx.h"
#include <iostream>
using namespace std;
int main()
{
	const int nrow = 2, ncol = 2;
	int b[nrow][ncol];
	int irow=-1, maxKol=0;
	cout << "Enter array elements\n" << nrow << " rows \n" << ncol << " columns\n";
	for (int i = 0; i < nrow; i++)
		for (int j = 0; j < ncol; j++)
			cin >> b[i][j];
	for (int i = 0; i < nrow; i++) {
		int count = 0;
		for (int j = 0; j < ncol; j++)
			if (b[i][j] == 0) count++;
		if (count > maxKol) { irow = i; maxKol = count; }
	}
```

```
	cout << "Initial array:\n";
	for (int i = 0; i < nrow; i++) {
		for (int j = 0; j < ncol; j++)
			cout << b[i][j];
		cout << "\n";
	}
	if (irow == -1) cout << "No zero rows\n";
	else cout << "max zero row is "<<irow<<endl;
	return 0;
}
```

Робота зі строками

Строка – це набір символів, який зберігається у послідовно розташованих комірках оперативної пам'яті.

В C++ підтримується два способи роботи зі строками.

1. Строки в стилі C.
2. Використання бібліотеки `string`.

Строки в стилі C

Кожний символ строки зберігається в масиві типу `char`. При цьому, останній символ повинен бути `'\0'`

```
char cat[8] = { 'f', 'a', ' t',  'e', 's', 's', 'a', '\0'}
char cat[] =”fat essa”
```

Приклад 3. Введення строк з клавіатури

```
//example 4_3.cpp Введення строк з клавіатури
//Приклад 3 до лабораторної роботи №4
//Автор: Кудін О.В., 01.10.2017
#include "stdafx.h"
#include <iostream>
#include <cstring>  // for the strlen() function
int main()
{
	using namespace std;
	const int Size = 15;
	char name1[Size];               // empty array
	char name2[Size] = "C++owboy";  // initialized array
                                                        // NOTE: some
implementations may require the static keyword
                                                        // to initialize the array
name2

	cout << "Howdy! I'm " << name2;
	cout << "! What's your name?\n";
	cin >> name1;
	cout << "Well, " << name1 << ", your name has ";
	cout << strlen(name1) << " letters and is stored\n";
	cout << "in an array of " << sizeof(name1) << " bytes.\n";
	cout << "Your initial is " << name1[0] << ".\n";
	name2[3] = '\0';                // set to null character
	cout << "Here are the first 3 characters of my name: ";
	cout << name2 << endl;
	return 0;
}
```

Тип даних `string`

В стандарті ISO/ANSI C++98 мова C++ була розширена додаванням типу даних `string` і відповідної бібліотеки `string`.

Приклад 4. Порівняне використання двох видів строк

```
//example 4_4.cpp Використання строкового типу
//Приклад 4 до лабораторної роботи №4
//Автор: Кудін О.В., 01.10.2017
#include "stdafx.h"
#include <iostream>
#include <string>               // make string class available
int main()
{
	using namespace std;
	char charr1[20];            // create an empty array
	char charr2[20] = "jaguar"; // create an initialized array
	string str1;                // create an empty string object
	string str2 = "panther";    // create an initialized string

	cout << "Enter a kind of feline: ";
	cin >> charr1;
	cout << "Enter another kind of feline: ";
	cin >> str1;                // use cin for input
	cout << "Here are some felines:\n";
	cout << charr1 << " " << charr2 << " "
	     << str1 << " " << str2 // use cout for output
	     << endl;
	cout << "The third letter in " << charr2 << " is "
	     << charr2[2] << endl;
	cout << "The third letter in " << str2 << " is "
	     << str2[2] << endl;    // use array notation

	return 0;
}
```

Операції зі строками

Одними з основних операцій над строками є присвоєння однієї строки іншій, конкатенація (додавання строки в кінець іншої).

Приклад 5. Операції зі строками.

```
//example 4_5.cpp Операції зі строками
//Приклад 5 до лабораторної роботи №4
//Автор: Кудін О.В., 01.10.2017
#include "stdafx.h"
#include <iostream>
#include <string>               // make string class available
int main()
{
	using namespace std;
	string s1 = "penguin";
	string s2, s3;

	cout << "You can assign one string object to another: s2 = s1\n";
	s2 = s1;
	cout << "s1: " << s1 << ", s2: " << s2 << endl;
	cout << "You can assign a C-style string to a string object.\n";
	cout << "s2 = \"buzzard\"\n";
	s2 = "buzzard";
	cout << "s2: " << s2 << endl;
	cout << "You can concatenate strings: s3 = s1 + s2\n";
	s3 = s1 + s2;
```

```
        cout << "s3: " << s3 << endl;
        cout << "You can append strings.\n";
        s1 += s2;
        cout << "s1 += s2 yields s1 = " << s1 << endl;
        s2 += " for a day";
        cout << "s2 += \" for a day\" yields s2 = " << s2 << endl;

        return 0;
}
```

Також, до основних функції роботи над строками є визначення довжини строки, та функції копіювання строк.

Приклад 6. Деякі функції бібліотеки `string`.

```
//example 4_6.cpp Функції над строками
//Приклад 6 до лабораторної роботи №4
//Автор: Кудін О.В., 01.10.2017
#include "stdafx.h"
#include <iostream>
#include <string>               // make string class available
#include <cstring>              // C-style string library
int main()
{
        using namespace std;
        char charr1[20];
        char charr2[20] = "jaguar";
        string str1;
        string str2 = "panther";

        // assignment for string objects and character arrays
        str1 = str2;                // copy str2 to str1
        strcpy(charr1, charr2);     // copy charr2 to charr1

        // appending for string objects and character arrays
        str1 += " paste";           // add paste to end of str1
        strcat(charr1, " juice");   // add juice to end of charr1

        // finding the length of a string object and a C-style string
        int len1 = str1.size();     // obtain length of str1
        int len2 = strlen(charr1);  // obtain length of charr1

        cout << "The string " << str1 << " contains "
                << len1 << " characters.\n";
        cout << "The string " << charr1 << " contains "
                << len2 << " characters.\n";
        return 0;
}
```

Завдання до лабораторної роботи

Розробити алгоритм (у вигляді блок-схеми) та програму для наведених нижче задач. Програма повинна задовольняти наступним вимогам:

- Передбачити введення даних з клавіатури та виведення результатів на екран консолі.
- Якщо потрібно, передбачити діалогове меню для вибору потрібної дії.
- Для кожної окремої дії програми передбачити окрему функцію.

- Обов’язковим є використання коментарів (відповідно до прикладів 1-5) та правил вибору ідентифікаторів в програмі (див. лаб. №2).
- Всі числові матриці та строки вводяться користувачем з клавіатури. Розмірності числових матриці задаються в коді програми.

1. Реалізувати додавання двох матриць довільної розмірності.
2. Реалізувати множення двох матриць довільної розмірності, якщо розмірність дозволяє операцію множення.
3. Для введених матриць другого або третього порядку обчислити визначник.
4. Знайти суму всіх чисел вище головної діагоналі матриці.
5. Знайти та вивести на екран максимальне число кожної строки матриці та мінімальне число кожного стовпця.
6. Запросити в користувача ввести алгебраїчний вираз. Перевірити коректність числа відкриваючих та закриваючих дужок.
7. Для введеної користувачем строки перевірити, чи є ця строка паліндромом (паліндром – це строка, яка читається однаково вперед і назад).
8. Символи введеної користувачем строки перевести в верхній регістр.

Контрольні запитання

1. Дайте визначення поняття масив. Чим масив відрізняється від простого типу даних?
2. Як розташовані елементи масиву в оперативній пам’яті?
3. Як здійснюється ініціалізація масивів?
4. Способи оголошення та ініціалізації строк в С++?
5. Типові операції та функції над строками?