

ЛАБОРАТОРНАЯ РАБОТА № 3

ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ

1. ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Целью работы является получение обучающимися навыков работы со средствами обработки одномерных массивов в языке C++.

Задачами работы являются:

- написание программ на языке C++ с использованием одномерных массивов;
- тестирование работоспособности программы для различных исходных данных.

2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Основная цель любой программы состоит в обработке данных. Данные различного типа хранятся и обрабатываются по-разному. В любом алгоритмическом языке каждая константа, переменная, результат вычисления выражения или функции должны иметь определенный тип.

Все типы языка C++ можно разделить на основные и составные. В языке C++ определено пять основных типов данных для представления целых, вещественных (`double` и `float`), символьных и логических величин. На основе этих типов программист может вводить описание составных типов. К ним относятся массивы, перечисления, функции, структуры, ссылки, указатели, объединения и классы.

При обработке данных достаточно часто приходится работать с рядом переменных одинакового типа (и описывающих одинаковые объекты). В этом случае эти переменные имеет смысл объединить одним идентификатором. Это позволяют сделать массивы.

Массив – это поименованный набор однотипных компонентов (элементов массива), расположенных в памяти непосредственно друг за другом.

Массивы обладают рядом достоинств:

- лёгкость вычисления адреса элемента по его индексу (поскольку элементы массива располагаются один за другим);
- одинаковое время доступа ко всем элементам;
- малый размер элементов: они состоят только из информационного поля.

Из объявления массива компилятор должен получить информацию о типе элементов массива и их количестве. Каждый элемент массива определяется именем массива и порядковым номером элемента, который называется индексом. Индекс в языке C++ – всегда целое число. Индекс в языке C++ указывается в квадратных скобках после имени массива. Количество используемых индексов в массиве определяет его размерность. Если в массиве один индекс, то это одномерный массив. Основная форма объявления (описания) одномерного массива имеет вид:

тип имя_массива [размер];

где тип – тип элементов массива,

размер – количество элементов одномерного массива.

Размер массива в языке C++ может задаваться константой или выражением, состоящим из констант. Нельзя задавать массив переменного размера. Для этого существует отдельный механизм, называемый динамическим выделением памяти.

Примеры описания массивов:

```
const int N=15;
```

```
double a[10], arr[N];
```

В языке C++ индекс всегда начинается с нуля, под первым элементом массива подразумевается элемент с индексом 0.

Если объявлен массив:

```
int x[100];
```

то это означает, что массив содержит 100 элементов от $x[0]$ до $x[99]$. Для одномерного массива легко подсчитать, сколько байт в памяти будет занимать этот массив:

колич.байт = размер_типа_элемента*колич.элементов

В языке C++ под массив всегда выделяется непрерывное место в оперативной памяти.

В языке C++ не проверяется выход индекса за пределы массива. Если массив $x[100]$ описан как целочисленный массив, имеющий 100 элементов, а в программе указан элемент $x[200]$, то сообщение об ошибке не будет выдано, а в качестве значения элемента $x[200]$ будет выдано некоторое число, занимающее 2 байта.

Используя циклы, можно осуществить перебор всего массива и через индексы обратиться к его элементам. Наиболее пригодными для решения этой задачи являются циклы со счетчиком.

Пример 1. Осуществить ввод элементов целочисленного массива с клавиатуры. Вывести значения элементов массива на экран в строку через пробел.

Программа на языке C++.

```
#include <locale.h>
int main( )
{
    setlocale(0, "");
    const int N=10; // размер массива
    double a[N]; // описание массива
    int i; // переменная-счетчик
    cout << "Введите " << N << " чисел" << endl;
    for (i=0; i<N; i++)
        cin << a[i]; // ввод элементов массива
    cout << "Массив" << endl; // вывод заголовка
    for (i=0; i<N; i++)
        cout << a[i] << " "; // // вывод элементов
    cout << endl;
}
```

В программе продемонстрировано использование при выводе символов русского алфавита (кириллицы). Для этого подключен заголовочный файл локализации `locale.h`, который содержит функции и классы для потоковой обработки данных в форме естественной для разных языков (денежный формат, представление символов, сортировка строк). Локаль содержит информацию о том, как интерпретировать и выполнять определенные операции ввода/вывода и преобразования с учетом специфики языков в определённых условиях.

Функция `setlocale` задает локаль, которая будет использоваться текущей программой. Можно изменить все параметры локали, или конкретные её части. Эта функция также может быть использована для получения имени текущей локали, передав `NULL` в через параметр `locale`.

Использование русского алфавита осуществляется вызовом функции `setlocale(LC_ALL, "Russian")`, можно опустить последний аргумент функции и заменить на `setlocale(0, "")`.

Элементы массива могут задаваться в зависимости от условий задачи различными способами:

- вводиться с клавиатуры;
- рассчитываться по формуле;
- генерироваться из некоторого диапазона.

Случайные числа в [языке программирования C++](#) могут быть сгенерированы функцией `rand()` из стандартной библиотеки C++. Функция `rand()` генерирует числа в диапазоне от 0 до `RAND_MAX`. `RAND_MAX` – это константа, определённая в библиотеке `<stdlib.h>`.

Зачастую, не нужен такой большой диапазон чисел от 0 до `RAND_MAX`. Например, в игре «Наперстки» необходимо отгадать, под каким из трёх напёрстков спрятан шарик, то есть генерация чисел должна выполняться в пределе от 1 до 3-х. Бросая монету, может возникнуть только два случая, когда монета упадёт «орлом» или «решкой» вверх, нужный интервал — от 1 до 2. Для того чтобы масштабировать интервал генерации чисел нужно воспользоваться, операцией нахождения остатка от деления «%». Например, какое бы не выдал число генератор случайных чисел `rand()`, запись `rand() % 3` в итоге выдаст число из диапазона от 0 до 2. Для того чтобы сместить диапазон, прибавляется единица: `rand() % 3 + 1` тогда диапазон изменится от 1 до 3 включительно.

При повторном запуске программы, печатаются те же самые числа. Суть в том, что функция `rand()` один раз генерирует случайные числа, а при последующих запусках программы всего лишь отображает сгенерированные первый раз числа. Такая особенность функции `rand()` нужна для того, чтобы можно было правильно отладить разрабатываемую программу. При отладке программы, внося какие-то изменения, необходимо удостовериться, что программа срабатывает правильно, а это возможно, если входные данные остались те же, то есть сгенерированные числа. Когда программа успешно отлажена, нужно, чтобы при каждом выполнении программы генерировались случайные числа. чтобы производить рандомизацию автоматически, то есть, не меняя каждый раз аргумент в функции `srand()` нужно воспользоваться функцией `time()` с аргументом 0. Чтобы использовать функцию `time()`, необходимо подключить заголовочный файл `<time.h>`.

Пример 2. Заполнить массив 20 случайными целыми числами из диапазона от 0 до 99.

Вычислить сумму его элементов.

Программа на языке C++.

```
#include "stdafx.h"
#include <iostream>
#include <time.h>
#include <stdlib.h>
#include <locale.h>

using namespace std;

int main()
{
    setlocale(0, "");
    const int n=20;
    int i, Array[n], Summ;
    srand(time(NULL));
    // генерация элементов массива
    for (i = 0; i < n; i++)
        Array[i] = rand() % 100;
    // Вывод элементов массива
    cout << "Сгенерированный массив" << endl;
    for ( i = 0; i < n; i++)
        cout << Array[i]<<" ";
    cout << endl;
    // вычисление суммы и вывод результата на экран
    Summ = 0;
    for ( i = 0; i < n; i++)
        Summ + = Array[i];
    cout <<"Summ = " << Summ <<endl;
    system("pause");
    return 0;
}
```

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Получение варианта задания на лабораторную работу.
2. Разработка алгоритма и графической схемы решения задачи.
3. Составление программы на языке C++.
4. Отладка программы.
5. Тестирование программы.
6. Составление отчета о проделанной работе.

4. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

1. Оформить титульный лист с указанием темы работы.
2. Сформулировать цель и задачи работы.
3. Привести формулировку задания.
4. Построить графическую схему алгоритма.
5. Записать текст программы.
6. Привести результаты тестирования программы.
7. Сформулировать вывод по проделанной работе.

5. ВАРИАНТЫ ЗАДАНИЙ

1. Дан целочисленный массив A размера N . Вывести номера тех его элементов $A[i]$, которые удовлетворяют двойному неравенству: $A[0] < A[i] < A[N-1]$. Если таких элементов нет, то вывести сообщение. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; 10k]$, где k – номер студента в списке группы.
2. Дан целочисленный массив размера N . Преобразовать его, прибавив к четным числам первый элемент. Первый и последний элементы массива не изменять. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-10k; 5k]$, где k – номер студента в списке группы.
3. Дан целочисленный массив размера N . Вывести сначала все его элементы, являющиеся четными числами, а затем – нечетными. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-5k; 2k]$, где k – номер студента в списке группы.

4. Поменять местами минимальный и максимальный элементы одномерного массива размера N . Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-6k; 0]$, где k – номер студента в списке группы.
5. Заменить все отрицательные элементы целочисленного массива размера N на минимальное значение элементов массива. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-3k; 4k]$, где k – номер студента в списке группы.
6. Дан массив размера N . Найти среднее арифметическое отрицательных элементов массива. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; 20k]$, где k – номер студента в списке группы.
7. Проверить, образуют ли элементы целочисленного массива размера N арифметическую прогрессию. Если да, то вывести разность прогрессии, если нет – вывести сообщение. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-k; 3k]$, где k – номер студента в списке группы.
8. Дан массив ненулевых целых чисел размера N . Проверить, чередуются ли в нем положительные и отрицательные числа. Если чередуются, то вывести сообщение, если нет, то вывести номер первого элемента, нарушающего закономерность. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-5k; 5k]$, где k – номер студента в списке группы.
9. Даны два одномерных массива a и b одинакового размера N . Найти $\max(a[i]-b[i])$, $0 \leq i < N$. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; 20k]$, где k – номер студента в списке группы.
10. Расположить элементы данного массива в обратном порядке (первый элемент меняется с последним, второй – с предпоследним и т.д. до среднего элемента). Провести тестирование программы при пользовательском вводе элементов

- массива и при генерации элементов массива из диапазона $[-3k; 0]$, где k – номер студента в списке группы.
11. В заданном одномерном массиве максимальный элемент заменить значением суммы предшествующих ему элементов. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; 10k]$, где k – номер студента в списке группы.
 12. В данном массиве поменять местами элементы, стоящие на нечетных местах, с элементами, стоящими на четных местах. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-2k; 5k]$, где k – номер студента в списке группы.
 13. Найти сумму нечетных элементов, стоящих на нечетных местах (то есть имеющих нечетные номера). Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-4k; 2k]$, где k – номер студента в списке группы.
 14. Дан одномерный целочисленный массив. Необходимо “сжать” массив, удалив из него каждый второй элемент, а оставшиеся элементы заменить нулями. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-k; 0]$, где k – номер студента в списке группы.
 15. Дан целочисленный массив и числа a и b (вводятся пользователем). Переместить в начало массива элементы, находящиеся в промежутке $[a, b]$. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-3k; 4k]$, где k – номер студента в списке группы.
 16. Задан массив с количеством элементов N . Сформируйте два массива: в первый включите элементы исходного массива с четными номерами, а во второй – с нечетными. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-k; 3k]$, где k – номер студента в списке группы.
 17. Дана последовательность N целых чисел и целое число m . Указать пары чисел этой последовательности таких, что их сумма равна m . Провести тестирование программы

при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; k]$, где k – номер студента в списке группы.

18. Дано N чисел. Наименьший член этой последовательности заменить значением среднего арифметического всех элементов, остальные элементы оставить без изменения. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-2k; 2k]$, где k – номер студента в списке группы.
19. Дана последовательность из N различных целых чисел. Найти сумму ее членов, расположенных между максимальным и минимальным значениями (в сумму включить и оба этих числа). Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; 100k]$, где k – номер студента в списке группы.
20. Даны два массива a и b одинаковой размерности N . Найти $\min |a[i] - b[i]|$, $0 < i < N$. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-k; 5k]$, где k – номер студента в списке группы.
21. Найти сумму всех четных элементов массива, стоящих на четных местах, то есть имеющих четные номера. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; 10k]$, где k – номер студента в списке группы.
22. В одномерном массиве заменить нулями все положительные элементы, идущие после минимального элемента массива. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-4k; 4k]$, где k – номер студента в списке группы.
23. В заданном массиве определить среднее арифметическое значение положительных и среднее арифметическое значение отрицательных элементов. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-2k; 3k]$, где k – номер студента в списке группы.
24. Найти среднее геометрическое положительных элементов одномерного вещественного массива. Провести тестирование

программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[-2k; 2k]$, где k – номер студента в списке группы.

25. Даны два одномерных целочисленных массива, состоящие из одинакового числа элементов. Получить третий массив той же размерности, каждый элемент которого равен большему из соответствующих элементов данных массивов. Провести тестирование программы при пользовательском вводе элементов массива и при генерации элементов массива из диапазона $[0; 3k]$, где k – номер студента в списке группы.

6. КРИТЕРИИ РЕЗУЛЬТАТИВНОСТИ ВЫПОЛНЕНИЯ РАБОТЫ

Лабораторная работа считается успешно выполненной, если:

- представленный отчет содержит все необходимые пункты, согласно требованиям;
- успешно демонстрируется работа программы как на выбранных студентом исходных данных, так и на заданных преподавателем исходных данных;
- студент правильно отвечает на контрольные вопросы.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое массив
2. Какие виды массивов существуют?
3. Как описывается массив?
4. Каких типов могут быть элементы массива?
5. Какими способами можно обращаться к элементам массива?
6. Каково назначение индекса и какие значения он может принимать на языке C++?
7. Какое начальное значение имеет индекс массива?
8. Что произойдет при выходе индекса массива за пределы указанного размера?
9. Как заполнить массив с помощью генератора случайных чисел?
10. Как выводить символы русского алфавита?