Лабораторная работа №5

Перегрузка функций.

ЗАДАНИЕ

Вариант 6

1. Для любого своего задания из лабораторных\_работ о массивах (№ 5, 6 или 7 прошлого семестра) реализовать обработку массива с использованием функций.
   1. Реализовать перегрузку функций для массивов целого, вещественного, символьного типа (там, где это возможно).
   2. Протестировать программу на вызов функций со всеми типами параметров.

В одномерном массиве, сформированном случайным образом, подсчитать количество элементов равных заданному числу х (вводится с клавиатуры).

ХОД РАБОТЫ

ЗАДАНИЕ 1

Результат работы программы представлен в соответствии с рисунком 1.

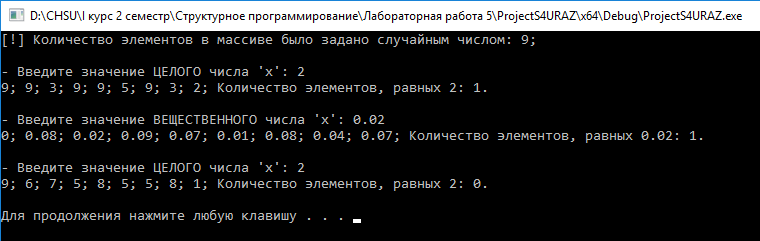


Рисунок 1 – результаты работы программы.

Текст программы:

#include <iostream>

#include <string>

#include <time.h>

using namespace std;

void minput(int n, int\* a) {

for (int i = 0; i < n; i++) {

a[i] = rand() % 10;

}

}

void minput(int n, float\* a) {

for (int i = 0; i < n; i++) {

a[i] = (float) (rand() % 10)/100;

}

}

void minput(int n, char\* a) {

for (int i = 0; i < n; i++) {

a[i] = rand() % 10;

}

}

void moutput(int n, int\* a) {

for (int i = 0; i < n; i++) {

cout << a[i] << "; ";

}

}

void moutput(int n, float\* a) {

for (int i = 0; i < n; i++) {

cout << a[i] << "; ";

}

}

void moutput(int n, char\* a) {

for (int i = 0; i < n; i++) {

cout << (int) a[i] << "; ";

}

}

int kvoid(int n, int\* a, int x) {

int k = 0;

for (int i = 0; i < n; i++) {

if (a[i] == x)

k++;

}

return k;

}

float kvoid(int n, float\* a, float x) {

float k = 0;

for (int i = 0; i < n; i++) {

if (a[i] == x)

k++;

}

return k;

}

int kvoid(int n, char\* a, int x) {

int k = 0;

for (int i = 0; i < n; i++) {

if ((int) a[i] == x)

k++;

}

return k;

}

int random() {

srand(time(0));

int r = rand() % 10;

return r;

}

void main() {

setlocale(LC\_CTYPE, "rus");

int n = random();

while (n < 1) {

n = random();

}

cout << "[!] Количество элементов в массиве было задано случайным числом: " << n << ";\n\n";

int x, k = 0;

cout << "- Введите значение ЦЕЛОГО числа 'х': ";

cin >> x;

int\* a = new int[n];

float\* b = new float[n];

char\* c = new char[n];

minput(n, a);

moutput(n, a);

cout << "Количество элементов, равных " << x << ": " << kvoid(n, a, x) << "." << endl;

cout << endl << "- Введите значение ВЕЩЕСТВЕННОГО числа 'х': ";

float x2;

cin >> x2;

minput(n, b);

moutput(n, b);

cout << "Количество элементов, равных " << x2 << ": " << kvoid(n, b, x2) << "." << endl;

cout << endl << "- Введите значение ЦЕЛОГО числа 'х': ";

cin >> x;

minput(n, c);

moutput(n, c);

cout << "Количество элементов, равных " << x << ": " << kvoid(n, c, x) << "." << endl;

cout << endl;

system("pause");

}

Алгоритм программы:

Объявляется функция minput с двумя параметрами: размер массива, указатель на первый элемент целочисленного массива a. Функция заполняет массив случайными значениями. Аналогично для вещественного массива b объявляется функция minput, только вторым параметром является указатель на первый элемент массива с вещественным типом данных float, а сами элементы заполняются случайными значениями, преобразованные в вещественный тип данных. Для символьного массива c необходимо, чтобы вторым параметром функции являлся указатель на первый элемент массива с типом данных char.

Объявляется функция moutput с двумя параметрами: размер массива, указатель на первый элемент массива с соответствующим типом данных. Функция выводит все элементы массива. Для вывода массива с символьным типом данных необходимо преобразовать тип данных в int.

Объявляется функция kvoid с тремя параметрами: размер массива, указатель на первый элемент массива и число х. Функция сравнивает каждый элемент массива с числом х, и, если они равны, то прибавляет к счётчику единицу и выводит значение счётчика в результате обращения к функции.

Пользователь задаёт значение числа х на клавиатуре. Объявляется целочисленный, вещественный и символьный одномерные динамические массивы a, b, c размером n элементов. Вызывается функция minput и moutput с соответствующими параметрами. Вызывается функция kvoid и выводится на экран результат вызова функции.

Вывод:

В ходе выполнения лабораторной работы мы научились использовать перегрузку функций для массивов с различными типами данных и закрепили навыки обработки массива с использованием функций.