Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

Дисциплина: Основы алгоритмизации и программирования

Тема: Перегрузка функций, функции с переменным числом параметров

|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Деревнин И.В. | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

**Анализ предметной области**

**Постановка задачи**

1. Создать функцию и перегрузить ее:

а) для массива целых чисел удаляет все четные элементы из массива;

б) для строки удаляет все четные слова.

2. Написать функцию (или макроопределение), которая находит длину стороны по координатам его точек. Написать функцию square, которая вычисляет площадь треугольника, заданного координатами вершин. Написать функцию squaren c переменным числом параметров, которая определяет площадь выпуклого многоугольника, заданного координатами своих вершин.

**Анализ задачи**

1. Дан динамический массив arr для массива целых чисел и text для строки. Массив целых чисел заполняется случайными числами, массив строк заполняется в тексте программы. Функция Delete перегружена для строк и целых чисел и удаляет четные элементы из массивов. Так как часть программы использует строки как формат данных, то необходимо использовать библиотеку string.

Для того чтобы протестировать программу необходимо:

а. протестировать работу функции Delete для четного и нечетного числа элементов с целочисленным значениями.

б. протестировать работу функции Delete для четного и нечетного числа элементов со строковым форматом данных

1. Функция length вычисляет длину вектора по координатам его точек. Функция square вычисляет площадь треугольника по координатам его точек, при этом используется функция length, которая вызывается рекурсивно. При вычислениях используется формула Герона. Функция squaren – функция с переменным числом параметров, которая вычисляет площадь выпуклого n-угольника исходя из формулы Гаусса (формулы шнурков). Так как параметры передаются в нее парами, то переключение параметров происходит через один. Так как программа основана на математических вычислениях, то необходимо использовать библиотеку cmath.

Для того чтобы протестировать программу необходимо:

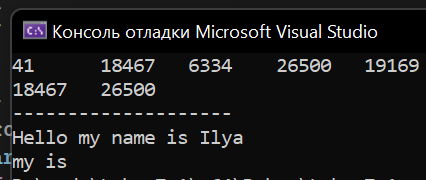
а. Протестировать корректность вычисления функции length.

б. Протестировать корректность вычисления функции square.

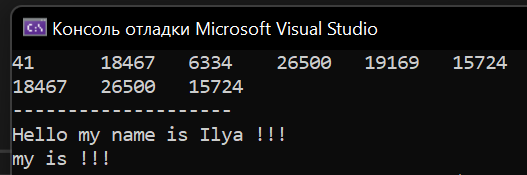
в. Протестировать корректность вычисления функции squaren.

**Тестирование программ**

1. Для нечетного количества элементов в массивах с числами и строками:

*Рис.* * 1 – Тестирование программы 1 для нечетного количества элементов.*

Для четного количества элементов:

*Рис. 2 - Тестирование программы 1 для четного количества элементов.*

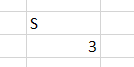
1. Тестирование функции length программы 2.

*Рис 3. – Тестирование функции length.*

*Рис. 4 – Истинное значение длины.*

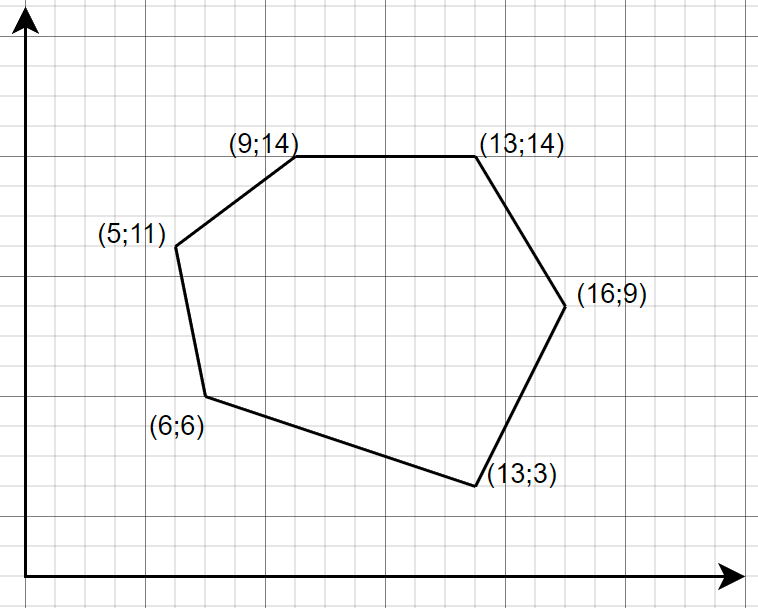
Тестирование функции square программы 2.

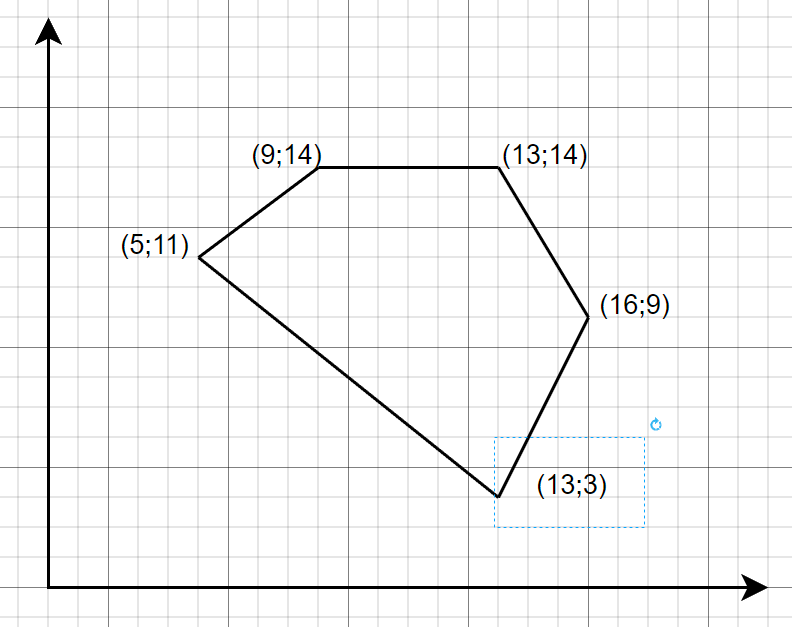
*Рис. 5 – Тестирование функции square.*

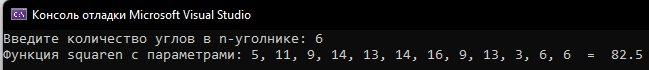
*Рис. 6 – Истинное значение площади треугольника.*

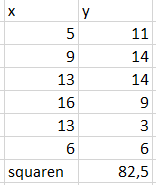
Тестирование функции squaren программы 2.

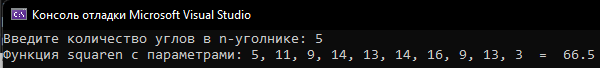
Для тестирования данной функции я выбрал 2 правильных многоугольника:

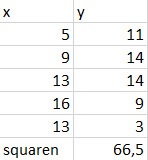
*Рис. 7 – Первый многоугольник.*

*Рис. 8 – Второй многоугольник.*

*Рис. 9 – Тестирование функции squaren для 1-го многоугольника.*

*Рис. 10 – Истинное значение площади первого n-угольника.*

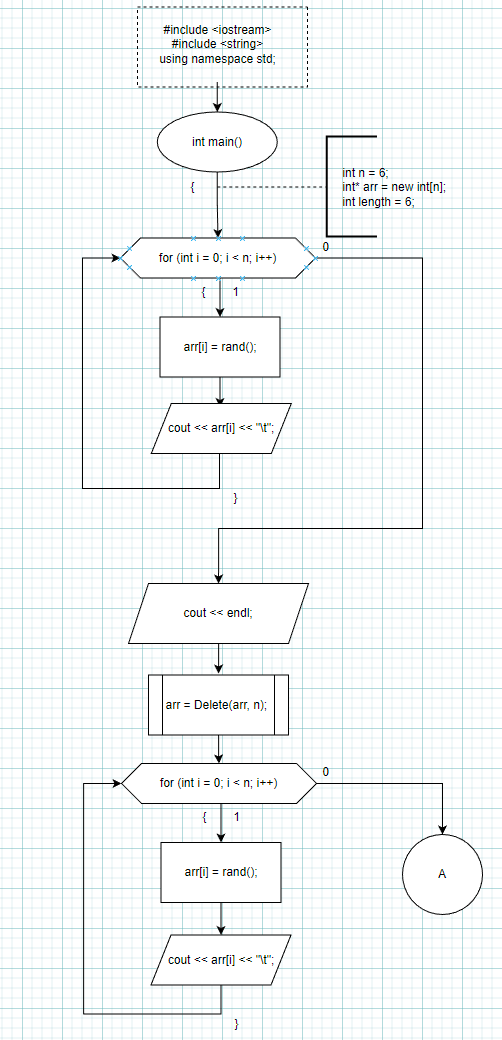
*Рис. 11 – Тестирование функции squaren для 2-го многоугольника.*

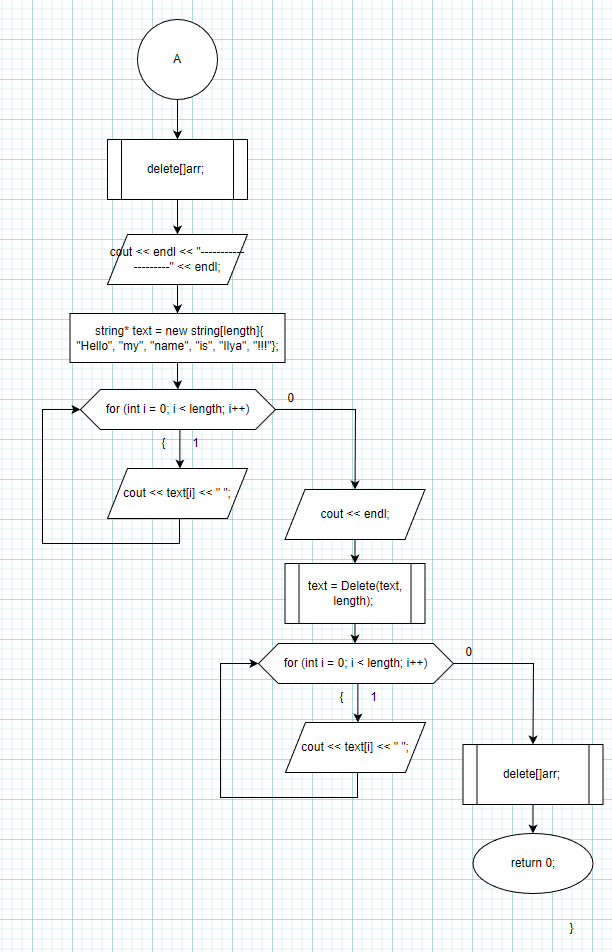
*Рис. 12 – Истинное значение площади второго n-угольника.*

**Заключение**

1. Была разработана программа, в которой перегружены функция Delete. Программа удаляет все четные элементы из массива, которые передаются в функцию. Программа работает исправно как для четного, так и для нечетного количества элементов в массиве. Но данная функция хоть и перегружена, но не совершенна, так как логика у функций абсолютно одинакова, то в данном случае разумно использовать не перегрузку функций, а шаблонную функцию, таким образом мы сможем обрабатывать любые типы данных, даже пользовательские.
2. Была разработана программа, которая стабильно находит длину отрезка заданного 2-мя точками на плоскости (функция length), площадь треугольника заданного 3-мя точками (функция square), а также площадь выпуклого многоугольника (функция squaren). Программа работает стабильно.

**Приложения**

Приложение А.1 – блок схема программы 1. 



Приложение Б.1 – код программы 1.

#include <iostream>

#include <string>

using namespace std;

int\* Delete(int\* arr, int& size) //Функция удаления четных элементов из массива целых чисел.

{

size /= 2; //Уменьшение длины массива в 2 раза

int\* new\_arr = new int[size]; //Создание нового динамического массива.

for (int i = 0; i < size; i++) //Выбор четных элементов

{

new\_arr[i] = arr[2 \* i + 1];

}

delete[]arr; //Очищение динамической памяти из под старого массива.

return new\_arr; //Возвращение нового указателя на массив

}

string\* Delete(string\* str, int &size) //Функция удаления четных слов из строки.

{

size /= 2; //Логика полностью аналогична логике с массивом целых чисел.

string\* new\_txt = new string[size];

for (int i = 0; i < size; i++)

{

new\_txt[i] = str[2 \* i + 1];

}

delete[]str;

return new\_txt;

}

int main()

{

int n = 6;

int\* arr = new int[n]; //Создание динамического массива для целых чисел

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

{

arr[i] = rand(); //заполнение рандомными элементами

cout << arr[i] << "\t";

}

cout << endl;

arr = Delete(arr, n); //вызов функции

for (int i = 0; i < n; i++) //вывод

{

cout << arr[i] << "\t";

}

delete[]arr; //очищение памяти из под динамического массива

cout << endl << "--------------------" << endl;

int length = 6;

string\* text = new string[length]{ "Hello", "my", "name", "is", "Ilya", "!!!"}; //создание строки, сразу с заполнением

for (int i = 0; i < length; i++) //вывод

{

cout << text[i] << " ";

}

cout << endl;

text = Delete(text, length); //вызов функции

for (int i = 0; i < length; i++) //вывод

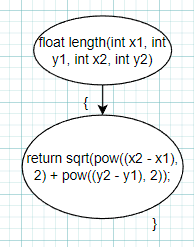
{

cout << text[i] << " ";

}

delete[]text; //очищение памяти из под динамического массива

return 0; }

Приложение А.2 – блок схема программы 2.

