

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «МОП ЭВМ»

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на
ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ
(09.03.02 Информационные системы и технологии)

Вариант 13

Студент группы БИСб-1

И. Икромов

Преподаватель

М.Е. Щелкунова

2018

Содержание

Содержание	2
Введение	4
1 Задание 7	5
1.1 Описание программы	5
1.2 Текст программы	5
1.3 Тестирование программы	6
2 Задание 8	7
2.1 Описание программы	7
2.2 Текст программы	7
2.3 Тестирование программы	8
3 Задание 10	9
3.1 Описание программы	9
3.2 Текст программы	9
3.3 Тестирование программы	10
4 Задание 11	11
4.1 Описание программы	11
4.2 Текст программы	11
4.3 Тестирование программы	12
5 Задание 12	13
5.1 Описание программы	13
5.2 Текст программы	13
5.3 Тестирование программы	14
6 Задание 13	15

6.1	Описание программы	15
6.2	Текст программы	15
6.3	Тестирование программы	16
	Заключение	17
	Список использованных источников	18

Введение

Язык C++ как средство обучения программированию обладает рядом несомненных достоинств. Он хорошо организован, строг, большинство его конструкций логичны и удобны. Развитые средства диагностики и редактирования кода делают процесс программирования приятным и эффективным.

Немаловажно, что C++ является не учебным, а профессиональным языком, предназначенным для решения широкого спектра задач, и в первую очередь - в быстро развивающейся области создания распределенных приложений.

1 Задание 7

1.1 Описание программы

Составить алгоритм для вычисления значения функции, заданной в виде графика, по введенному значению аргумента. Параметр R ввести с клавиатуры.

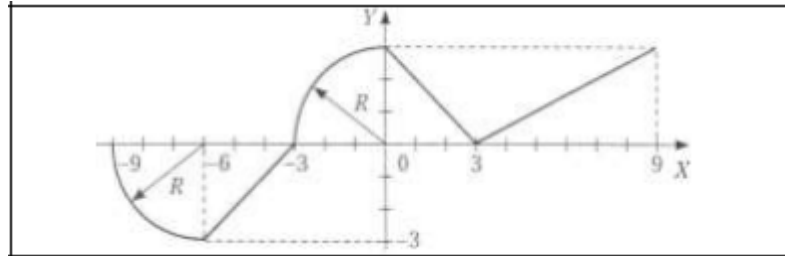


Рисунок 1.1 – График функции

1.2 Текст программы

Проект состоит из одного файла исходного кода, который приведен в листинге 1.1.

Листинг 1.1 – Текст файла prog1.cpp

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    float x, r, y;

    while (true)
    {
        cout << "Введите значение аргумента: ";
        cin >> x;
        cout << "Введите значение радиуса r: ";
        cin >> r;

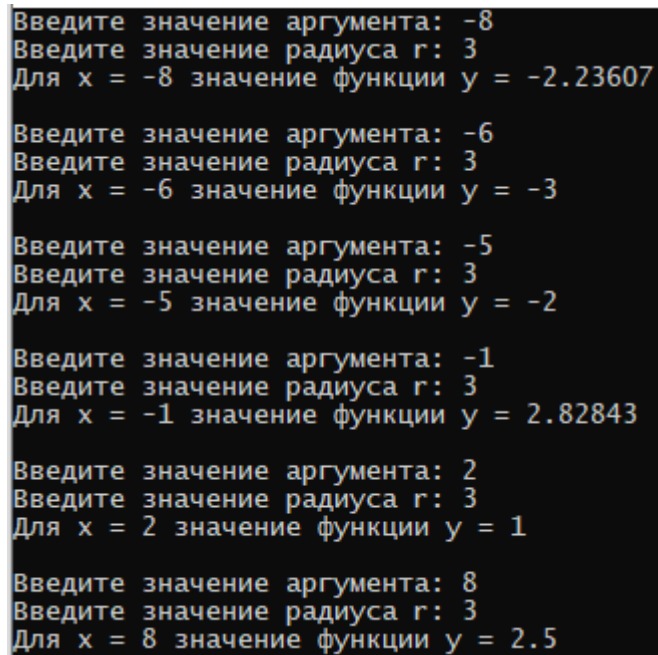
        if (x < -9 || x > 9)
        {
            cout << "Для x = " << x << " функция не определена" << endl;
        }
        else
        {
            if (x <= -6) y = sqrt(r * r - pow(x + 6, 2));
            else if (x <= -3) y = x + 3;
            else if (x <= 0) y = sqrt(r * r - pow(x, 2));
            else if (x <= 3) y = -x + 3;
            else y = (x - 3) / 2;
            cout << "Для x = " << x << " значение функции y = " << y << endl;
        }
    }

    cout << endl;
```

```
}  
  
    return 0;  
}
```

1.3 Тестирование программы

Результат работы программы приведен на рисунке 1.2.



```
Введите значение аргумента: -8  
Введите значение радиуса r: 3  
Для x = -8 значение функции y = -2.23607  
  
Введите значение аргумента: -6  
Введите значение радиуса r: 3  
Для x = -6 значение функции y = -3  
  
Введите значение аргумента: -5  
Введите значение радиуса r: 3  
Для x = -5 значение функции y = -2  
  
Введите значение аргумента: -1  
Введите значение радиуса r: 3  
Для x = -1 значение функции y = 2.82843  
  
Введите значение аргумента: 2  
Введите значение радиуса r: 3  
Для x = 2 значение функции y = 1  
  
Введите значение аргумента: 8  
Введите значение радиуса r: 3  
Для x = 8 значение функции y = 2.5
```

Рисунок 1.2 – Результат работы программы

2 Задание 8

2.1 Описание программы

Составить алгоритм для определения, попадает ли точка с произвольно заданными координатами (x, y) в область, закрашенную на рисунке серым цветом. Координаты точки (x, y) и другие необходимые данные задать самостоятельно.

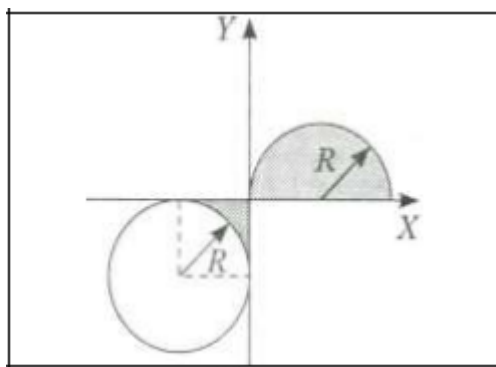


Рисунок 2.1 – График

2.2 Текст программы

Проект состоит из одного файла исходного кода, который приведен в листинге 2.1.

Листинг 2.1 – Текст файла prog2.cpp

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    float x, y, r;
    cout << "Введите значение радиуса окружности: ";
    cin >> r;

    while (true)
    {
        cout << "Координаты точки" << endl;
        cout << "Введите x: ";
        cin >> x;
        cout << "Введите y: ";
        cin >> y;

        if (
            (y >= 0 && x >= 0 && ((x - r) * (x - r) + y * y <= r * r))
            ||
            ((x <= 0 && y <= 0) && pow(x - (-r), 2) + pow(y - (-r), 2) >= r * r)
        )
        {
            // Point is in the shaded region
        }
    }
}
```

```

    {
        cout << "Точка попадает в область" << endl;
    }
    else
    {
        cout << "Точка не попадает в область" << endl;
    }
}

return 0;
}

```

2.3 Тестирование программы

Результат работы программы приведен на рисунке 2.2.

```

Введите значение радиуса окружности: 2
Координаты точки
Введите x: 2
Введите y: 2
Точка попадает в область
Координаты точки
Введите x: -1
Введите y: -1
Точка не попадает в область
Координаты точки
Введите x: -0.2
Введите y: -0.2
Точка попадает в область
Координаты точки
Введите x: 4
Введите y: 0
Точка попадает в область
Координаты точки
Введите x: 0
Введите y: -1
Точка попадает в область
Координаты точки
Введите x: 1
Введите y: 2
Точка не попадает в область
Координаты точки

```

Рисунок 2.2 – Результат работы программы

3 Задание 10

3.1 Описание программы

Разработать циклический алгоритм вычисления заданного выражения.

Дано натуральное число n , действительное число x . Вычислить:

$$\sum_{i=1}^n \frac{(x-2) \cdot (x-4) \cdot (x-8) \dots \cdot (x-2^i)}{(x-1) \cdot (x-3) \cdot (x-7) \dots \cdot (x-2^i+1)}.$$

Рисунок 3.1 – Ряд

3.2 Текст программы

Проект состоит из одного файла исходного кода, который приведен в листинге 3.1.

Листинг 3.1 – Текст файла prog3.cpp

```
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    int n;
    cout << "Введите натуральное число n: ";
    cin >> n;

    int x;
    cout << "Введите действительное число x: ";
    cin >> x;

    double res = 0, temp_rez, ch = 1, zn = 1;

    for (int i = 1; i <= n; i++)
    {
        // Вычисление числителя очередного члена ряда
        ch *= (x - pow(2,i));
        // Вычисление знаменателя очередного члена ряда
        zn *= (x - pow(2, i) + 1);
        // Вычисление очередного члена ряда
        temp_rez = ch / zn;
        res += temp_rez;

        cout << "Значение " << i << " члена ряда: " << temp_rez << endl;
    }
    cout << endl << "Сумма ряда: " << res << endl;

    system("pause");
    return 0;
}
```

3.3 Тестирование программы

Результат работы программы приведен на рисунке 3.1.

```
Введите натуральное число n: 5
Введите действительное число x: 18
Значение 1 члена ряда: 0.941176
Значение 2 члена ряда: 0.878431
Значение 3 члена ряда: 0.798574
Значение 4 члена ряда: 0.532383
Значение 5 члена ряда: 0.573335
Сумма ряда: 3.7239
```

Рисунок 3.2 – Результат работы программы

4 Задание 11

4.1 Описание программы

Разработать алгоритм для работы с цифрами натурального числа. Предусмотреть печать заданного числа и всех результатов.

Для заданного натурального числа k определить количество цифр кратных трем.

4.2 Текст программы

Проект состоит из одного файла исходного кода, который приведен в листинге 4.1.

Листинг 4.1 – Текст файла prog4.cpp

```
// Для заданного натурального числа k определить количество цифр кратных трем.
#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

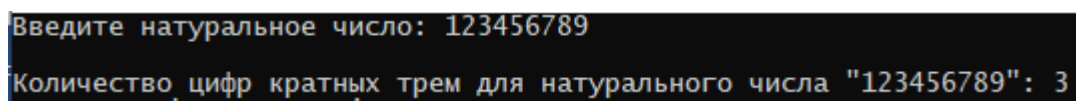
    int n, count = 0;
    cout << "Введите натуральное число: ";
    cin >> n;

    for (int n_temp = n; n_temp != 0; n_temp /= 10)
    {
        if (n_temp % 10 % 3 == 0)
        {
            count++;
        }
    }

    cout << endl;
    cout << "Количество цифр кратных трем для натурального числа \"<n>\" << n << "\": " <<
count << endl;
    system("pause");
    return 0;
}
```

4.3 Тестирование программы

Результат работы программы приведен на рисунке 4.1.



```
Введите натуральное число: 123456789
Количество цифр кратных трем для натурального числа "123456789": 3
```

Рисунок 4.1 – Результат работы программы

5 Задание 12

5.1 Описание программы

Разработать алгоритм обработки одномерных числовых массивов. Размер и значения элементов исходного массива задать самостоятельно, предусмотреть печать исходных данных и всех результатов.

Задана последовательность чисел d . Переписать в новый массив C нечетные числа заданного массива d , расположенные после последнего по порядку максимального числа.

5.2 Текст программы

Проект состоит из одного файла исходного кода, который приведен в листинге 5.1.

Листинг 5.1 – Текст файла prog5.cpp

```
/*
Переписать в новый массив C нечетные числа заданного массива A,
расположенные после последнего по порядку максимального числа
*/

#include <iostream>
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    int arr[] = {13, 11, -6, 13, 2, 5, 3, -12};
    const int arr_size = sizeof(arr) / sizeof(*arr);
    int arr_res[arr_size];
    int index_last_max_el = 0;

    // Нахождение последнего номера максимального элемента и распечатка массива
    cout << "Начальный массив: ";
    for (int i = 0; i < arr_size; i++)
    {
        cout << arr[i] << " ";
        if (arr[i] >= arr[index_last_max_el])
            index_last_max_el = i;
    }

    int counter = 0;
    // Заполнение нового массива нечетными числами расположенными
    // после последнего по порядку максимального числа исх массива
    for (int i = index_last_max_el + 1; i < arr_size; i++)
    {
        if (arr[i] % 2 != 0)
        {
            arr_res[counter] = arr[i];
            counter++;
        }
    }
}
```

```

        counter++;
    }
}

cout << endl;
cout << "Номер максимального элемента (счёт идёт с единицы): " << index_last_max_el+1
<< endl;
cout << "Новый массив заполненный нечетными числами расположенными после последнего
по порядку \nмаксимального элемента исходного массива: ";

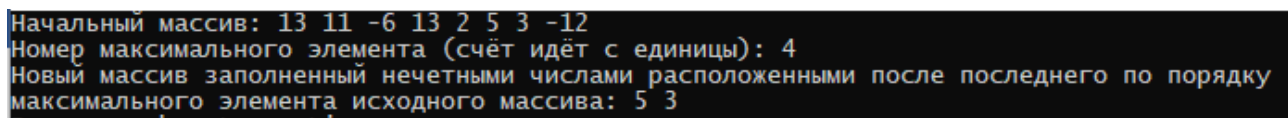
// Распечатка сформированного массива
for (int i = 0; i < counter; i++)
{
    cout << arr_res[i] << " ";
}
cout << endl;

system("pause");
return 0;
}

```

5.3 Тестирование программы

Результат работы программы приведен на рисунке 5.1.



```

Начальный массив: 13 11 -6 13 2 5 3 -12
Номер максимального элемента (счёт идёт с единицы): 4
Новый массив заполненный нечетными числами расположенными после последнего по порядку
максимального элемента исходного массива: 5 3

```

Рисунок 5.1 – Результат работы программы

6 Задание 13

6.1 Описание программы

Разработать алгоритм обработки одномерных числовых массивов. Размер и значения элементов исходного массива задать самостоятельно, предусмотреть печать значений элементов исходного и результирующего массивов.

Расширить заданную последовательность чисел С, вставив в нее заданное число А перед каждым числом, равным заданному числу В.

6.2 Текст программы

Проект состоит из одного файла исходного кода, который приведен в листинге 6.1.

Листинг 6.1 – Текст файла prog7.cpp

```
/*
Расширить заданную последовательность чисел с, вставив в нее заданное
число а перед каждым числом, равным заданному числу b
*/

#include <iostream>
using namespace std;

const int arr_max_size = 128;

int main()
{
    setlocale(LC_ALL, "Russian");

    int arr_current_size = 8;
    float arr[arr_max_size] = { 1, 5.2, -3, 13, -12.1, 1,6,1 };

    // Распечатка исходного массива
    cout << "Начальный массив: ";
    for (int i = 0; i < arr_current_size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;

    float el;
    cout << "Введите элемент для вставки: ";
    cin >> el;

    float before;
    cout << "Элемент исходного массива перед которым будет произведена вставка: ";
    cin >> before;

    // Вставка элемента в исходный массив
    for (int i = 0; i < arr_current_size; i++)
    {
```

```

        if (arr[i] == before)
        {
            // перемещение эл массива вправо на одну позицию
            for (int j = arr_current_size; j >= i; j--)
            {
                arr[j + 1] = arr[j];
            }
            arr[i] = e1;
            arr_current_size++;
            i+=2;
        }
    }

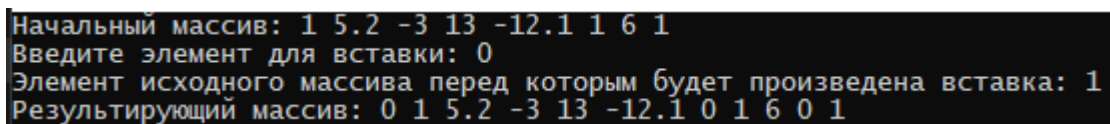
    cout << "Результирующий массив: ";
    // Распечатка результ массива
    for (int i = 0; i < arr_current_size; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;

    system("pause");
    return 0;
}

```

6.3 Тестирование программы

Результат работы программы приведён на рисунке 6.1.



```

Начальный массив: 1 5.2 -3 13 -12.1 1 6 1
Введите элемент для вставки: 0
Элемент исходного массива перед которым будет произведена вставка: 1
Результирующий массив: 0 1 5.2 -3 13 -12.1 0 1 6 0 1

```

Рисунок 6.1 – Результат работы программы

Заключение

В ходе прохождения летней производственной практики были получены знания по изучению языка программирования C++. Были рассмотрены такие темы как:

- 1 Линейные программы.
- 2 Условные операторы.
- 3 Разветвляющиеся вычислительные процессы.
- 4 Организация циклов.
- 5 Одномерные массивы.
- 6 Двумерные массивы.

Полученные навыки и знания будут использоваться в дальнейших проектах.

Список использованных источников

- 1 Павловская Т. А., С++ Программирование на языке высокого уровня: Практикум. — СПб.: Питер, 2009. — 432 с.: ил. — (Серия «Учебное пособие»).
- 2 Щелкунова, М. Е. Информатика: учеб. пособие / М. Е. Щелкунова — Комсомольск-на-Амуре: ФГБОУ ВПО «КнАГУ», 2012. — 100 с.