

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 15

Студент группы 6ИСб-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 ..... | 15 |
| 6   Задание 13 .....               | 16 |

|     |  |    |
|-----|--|----|
| 6.1 | Описание программы .....               | 16 |
| 6.2 | Текст программы .....                  | 16 |
| 6.3 | Тестирование программы .....           | 17 |
|     | Заключение .....                       | 18 |
|     | Список использованных источников ..... | 19 |

## **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.

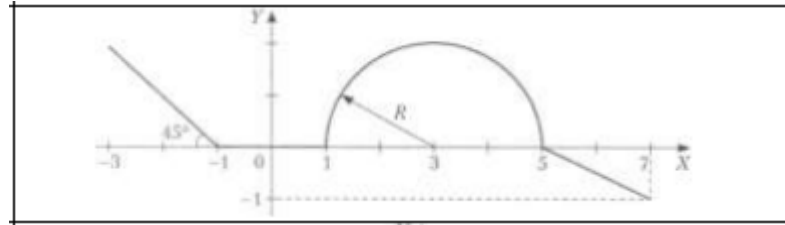


Рисунок 1.1 – График функции

### 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 < -3 || x > 7)
        {
            cout << "Для x = " << x << " функция не определена" << endl;
        }
        else
        {
            if (x <= -1) y = -x - 1;
            else if (x <= 1) y = 0;
            else if (x <= 5) y = sqrt(r*r - (x-3) * (x - 3));
            else y = -x / 2 + 2.5;
            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: 7
Введите значение радиуса r: 2
Для x = 7 значение функции y = -1

Введите значение аргумента: 6
Введите значение радиуса r: 2
Для x = 6 значение функции y = -0.5

Введите значение аргумента: 3
Введите значение радиуса r: 2
Для x = 3 значение функции y = 2

Введите значение аргумента: -2
Введите значение радиуса r: 2
Для x = -2 значение функции y = 1

Введите значение аргумента: -3
Введите значение радиуса r: 2
Для x = -3 значение функции y = 2
```

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.

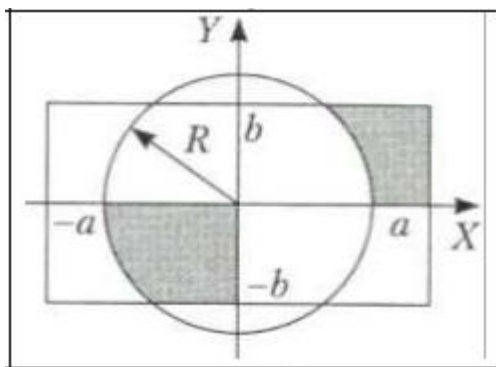


Рисунок 2.1 – График

### 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,height,width;
    cout << "Введите значение радиуса окружности: ";
    cin >> r;
    cout << "Введите высоту ограничивающего прямоугольника (не более " << r*2 <<"): ";
    cin >> height;
    cout << "Введите ширину ограничивающего прямоугольника (не менее " << r * 2 << "): ";
    cin >> width;

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

```

        if ((y <= sqrt(r - x * x) && y <= 0 && x <= 0 && y >= -(height / 2)) ||
            (y >= 0 && x >= 0 && y <= height / 2 && x <= width / 2 && (x * x + y
* y > r* r)))
        {
            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: 3
Введите высоту ограничивающего прямоугольника (не более 6): 4
Введите ширину ограничивающего прямоугольника (не менее 6): 8
Координаты точки
Введите x: -1
Введите y: -1
Точка попадает в область
Координаты точки
Введите x: -3
Введите y: -3
Точка не попадает в область
Координаты точки
Введите x: 0
Введите y: -3
Точка не попадает в область
Координаты точки
Введите x: 2
Введите y: 2
Точка не попадает в область
Координаты точки
Введите x: 4
Введите y: 1
Точка попадает в область

```

Рисунок 2.2 – Результат работы программы



### 3 Задание 10

#### 3.1 Описание программы

Разработать циклический алгоритм вычисления заданного выражения.

Дано натуральное число  $n$ . Вычислить:  $\sum_{i=1}^n 2^{i!}$ .

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");

    float n;
    cout << "Введите n: ";
    cin >> n;
    long long result = 0, tempRes = 1, temp;

    for (int i = 1; i < n; i++, tempRes=1)
    {
        // Вычисление степени очередного члена ряда
        for (int fact_temp = 1; fact_temp <= i; fact_temp++)
            tempRes *= fact_temp;

        // Вычисление очередного члена ряда
        tempRes = pow(2, tempRes);

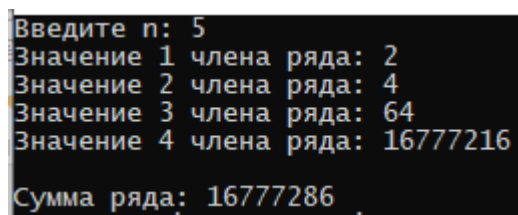
        // Прибавление очередного члена ряда к итоговой сумме
        result += tempRes;

        cout << "Значение " << i << " члена ряда: " << tempRes << endl;
    }
    cout << endl << "Сумма ряда: " << result << endl;

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

### 3.3 Тестирование программы

Результат работы программы приведен на рисунке 3.1.

A screenshot of a terminal window with a black background and yellow text. It displays the results of a program where the user has entered '5' for 'n'. The program has calculated the first four terms of a sequence and their sum.

```
Введите n: 5
Значение 1 члена ряда: 2
Значение 2 члена ряда: 4
Значение 3 члена ряда: 64
Значение 4 члена ряда: 16777216
Сумма ряда: 16777286
```

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.

Найти произведение цифр нечетных разрядов заданного натурального числа  $k$ .

### 4.2 Текст программы

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

Листинг 4.1 – Текст файла prog4.cpp

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

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

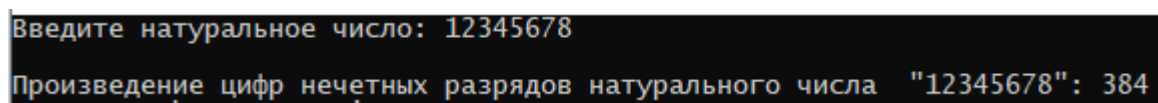
    int n, res = 1;
    cout << "Введите натуральное число: ";
    cin >> n;

    for (int n_temp = n, razr = 1; n_temp != 0; n_temp /= 10, razr++)
    {
        if (razr % 2 != 0)
        {
            res *= n_temp % 10;
        }
    }

    cout << endl;
    cout << "Произведение цифр нечетных разрядов натурального числа \'" << n << "\': "
    << res << endl;
    system("pause");
    return 0;
}
```

### 4.3 Тестирование программы

Результат работы программы приведен на рисунке 4.1.



```
Введите натуральное число: 12345678
Произведение цифр нечетных разрядов натурального числа "12345678": 384
```

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  $d$ . Найти в ней наибольший номер минимального значения. Сформировать новую последовательность  $d$  из нечетных чисел с четными номерами, расположенных до первого по порядку минимального числа в исходной последовательности  $d$ .

### 5.2 Текст программы

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

Листинг 5.1 – Текст файла prog5.cpp

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

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

    int arr[] = {1,4,2, 5, 3, 13, -12};
    int arr_sort[sizeof(arr) / sizeof(*arr)];
    int index_min_el = 0;

    // Нахождение номера минимального элемента и распечатка массива
    cout << "Начальный массив: ";
    for (int i = 0; i < sizeof(arr) / sizeof(*arr); i++)
    {
        cout << arr[i] << " ";
        if (arr[i] < arr[index_min_el])
            index_min_el = i;
    }

    int counter = 0;
    // Заполнение нового массива чётными эл
    for (int i = 1; i < sizeof(arr) / sizeof(*arr) + 1; i+=2)
    {
        if (arr[i] % 2 != 0)
        {
            arr_sort[counter] = arr[i];
            counter++;
        }
    }
}
```

```

        cout << endl;
        cout << "Номер минимального значения (счёт идёт с единицы): " << index_min_el+1 <<
endl;
        cout << "Последовательность из нечетных чисел с четными номерами, расположенных до
первого по порядку минимального числа в исходной последовательности: ";

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

        system("pause");
        return 0;
}

```

### 5.3 Тестирование программы

Результат работы программы приведен на рисунке 5.1.

```
Начальный массив: 1 4 2 5 3 13 -12  
Номер минимального значения (счёт идёт с единицы): 7  
Последовательность из нечетных чисел с четными номерами, расположенных до первого  
по порядку минимального числа в исходной последовательности: 5 13
```

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.

Преобразовать заданную последовательность чисел с путем удаления из нее всех положительных чисел с номерами от  $k$  до  $m$

### 6.2 Текст программы

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

Листинг 6.1 – Текст файла prog7.cpp

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

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

    float arr[] = { 1, 5.2, -3, 13, -12.1, 8, -5, 16, 48 };
    int arr_current_size = sizeof(arr)/sizeof(*arr);

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

    cout << "Диапазон положительных чисел для удаления (от 1 до " << arr_current_size <<
    "):" << endl;
    cout << "Начало диапазона: " << endl;
    int start;
    cin >> start;
    start--;
    cout << "Конец диапазона:" << endl;
    int end;
    cin >> end;
    end--;

    // Удаление элементов из исходного массива
    for (int i = 0; i < arr_current_size; i)
    {
        if (i >= start && i <= end && arr[i] > 0)
        {
            for (int j = i; j < arr_current_size - 1; j++)
            {
```



```

        arr[j] = arr[j + 1];
    }
    arr_current_size--;
    end--;
}
else i++;
}

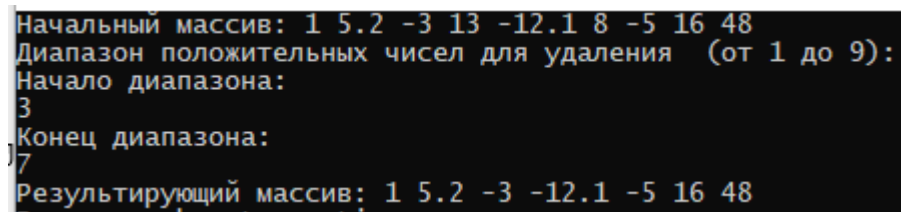
cout << "Результирующий массив: ";
// Распечатка результ массива
for (int i = 0; i < arr_current_size; i++)
{
    cout << arr[i] << " ";
}
cout << endl;

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 8 -5 16 48
Диапазон положительных чисел для удаления (от 1 до 9):
Начало диапазона:
3
Конец диапазона:
7
Результирующий массив: 1 5.2 -3 -12.1 -5 16 48

```

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.