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 3

Студент группы 6ИСб-1 И. Нозимзода

Преподаватель М.Е. Щелкунова

2019

Содержание

[Содержание 2](#_Toc18278977)

[Введение 4](#_Toc18278978)

[1 Задание 7 5](#_Toc18278979)

[1.1 Описание программы 5](#_Toc18278980)

[1.2 Текст программы 5](#_Toc18278981)

[1.3 Тестирование программы 6](#_Toc18278982)

[2 Задание 8 7](#_Toc18278983)

[2.1 Описание программы 7](#_Toc18278984)

[2.2 Текст программы 7](#_Toc18278985)

[2.3 Тестирование программы 8](#_Toc18278986)

[3 Задание 10 9](#_Toc18278987)

[3.1 Описание программы 9](#_Toc18278988)

[3.2 Текст программы 9](#_Toc18278989)

[3.3 Тестирование программы 10](#_Toc18278990)

[4 Задание 11 11](#_Toc18278991)

[4.1 Описание программы 11](#_Toc18278992)

[4.2 Текст программы 11](#_Toc18278993)

[4.3 Тестирование программы 12](#_Toc18278994)

[5 Задание 12 13](#_Toc18278995)

[5.1 Описание программы 13](#_Toc18278996)

[5.2 Текст программы 13](#_Toc18278997)

[5.3 Тестирование программы 14](#_Toc18278998)

[6 Задание 13 15](#_Toc18278999)

[6.1 Описание программы 15](#_Toc18279000)

[6.2 Текст программы 15](#_Toc18279001)

[6.3 Тестирование программы 16](#_Toc18279002)

[Заключение 17](#_Toc18279003)

[Список использованных источников 18](#_Toc18279004)

Введение

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

# Задание 7

## Описание программы

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

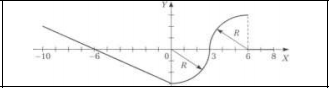


Рисунок 1.1 – График функции

## 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 < -10 || x > 8)  {  cout << "Для x = " << x << " функция не определенна" << endl;  }  else  {  if (x <= -6) y = (-x - 6)/2;  else if (x <= 0) y = -(x/2+3);  else if (x <= 3) y = (sqrt(r \* r - pow(x, 2)))\*(-1);  else if (x <= 6) y = sqrt(r \* r - pow(x - 6, 2)) ;  else y = 0;  cout << "Для x = " << x << " значение функции y = " << y << endl;  }  cout << endl;  }  return 0;  } |

## Тестирование программы

Результат работы программы приведен на рисунке 1.2.

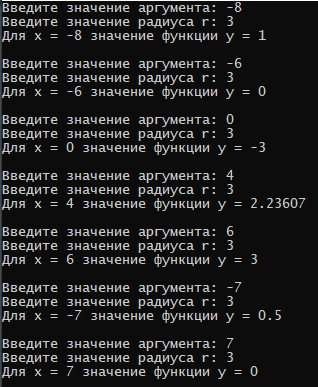


Рисунок 1.2 – Результат работы программы

# Задание 8

## Описание программы

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

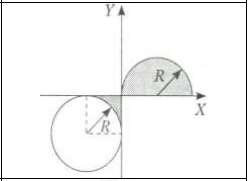


Рисунок 2.1 – График

## 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)  )  {  cout << "Точка попадает в область" << endl;  }  else  {  cout << "Точка не попадает в область" << endl;  }  }  return 0;  } |

## Тестирование программы

Результат работы программы приведен на рисунке 2.2.

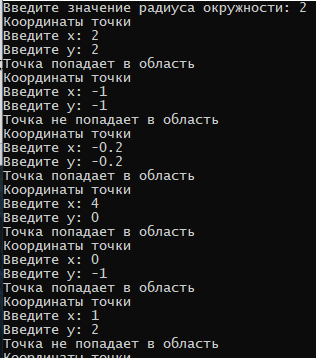


Рисунок 2.2 – Результат работы программы

# Задание 10

## Описание программы

Разработать циклический алгоритм вычисления заданного выражения.



Рисунок 3.1 – Ряд

## 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 a: ";  cin >> x;  double res = 0, temp\_rez, ch = 1, zn = 1;  for (int i = 0; i <= n; i++)  {  // Вычисление числителя очередного члена ряда  ch \*= (pow(-1,i));  // Вычисление знаменателя очередного члена ряда  zn \*= pow(x, 2 \* i);  // 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;  } |

## Тестирование программы

Результат работы программы приведен на рисунке 3.1.

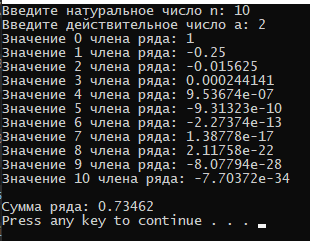


Рисунок 3.2 – Результат работы программы

# Задание 11

## Описание программы

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

Для заданного натурального числа k определить сумму цифр нечетных разрядов заданного натурального числа m.

## Текст программы

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

Листинг 4.1 – Текст файла prog4.cpp

|  |
| --- |
| // Определить сумму цифр нечетных разрядов заданного натурального числа m.  #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_ALL, "Russian");  int n,res = 0;  cout << "Введите натуральное число: ";  cin >> n;  // true - чётное количество разрядов, false - нечётное количество (0 это четное число)  bool evenNumber = true;  // Определяем чётность  for (int n\_temp = n; n\_temp != 0; n\_temp /= 10) evenNumber = !evenNumber;  // Находим сумму цифр нечётных разрядов  // Если число чётное начинаем считать с 1 разряда с конца, если нечётное со 0 разряда с конца  for (int n\_temp = n, razr = evenNumber; n\_temp != 0; n\_temp /= 10, razr++)  {  if (razr % 2 == 0)  {  res += n\_temp % 10;  }  }  cout << endl;  cout << "Cумма цифр нечетных разрядов заданного натурального числа \"" << n << "\": " << res << endl;  system("pause");  return 0;  } |

## Тестирование программы

Результат работы программы приведен на рисунке 4.1.

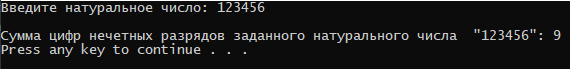


Рисунок 4.1 – Результат работы программы

# Задание 12

## 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 до первого по порядку минимального числа.

## Текст программы

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

Листинг 5.1 – Текст файла prog5.cpp

|  |
| --- |
| /\*  Нечетные числа заданного массива d, расположенные до первого по порядку минимального числа, переписать в новый массив b  \*/  #include <iostream>  using namespace std;  int main()  {  setlocale(LC\_ALL, "Russian");  int arr[] = {13, 11, -6, 13, 2, -12, 5, 3, -12};  const int arr\_size = sizeof(arr) / sizeof(\*arr);  int arr\_res[arr\_size];  int index\_last\_min\_el = 0;  // Нахождение первого номера минимального числа и распечатка массива  cout << "Начальный массив: ";  for (int i = 0; i < arr\_size; i++)  {  cout << arr[i] << " ";  if (arr[i] < arr[index\_last\_min\_el])  index\_last\_min\_el = i;  }  int counter = 0;  // Заполнение нового массива нечетными числами расположенными  // до первого по порядку максимального числа исх массива  for (int i = 0; i < index\_last\_min\_el; i++)  {  if (arr[i] % 2 != 0)  {  arr\_res[counter] = arr[i];  counter++;  }  }  cout << endl;  cout << "Номер первого минимально элемента (счёт идёт с единицы): " << index\_last\_min\_el+1 << endl;  cout << "Новый массив заполненный нечетными числами расположенными до первого \nминимального элемента исходного массива: ";  // Распечатка сформированного массива  for (int i = 0; i < counter; i++)  {  cout << arr\_res[i] << " ";  }  cout << endl;  system("pause");  return 0;  } |

## Тестирование программы

Результат работы программы приведен на рисунке 5.1.

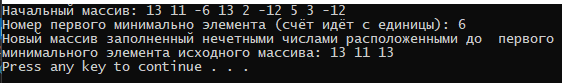


Рисунок 5.1 – Результат работы программы

# Задание 13

## 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 чисел больше заданного значения А.

## Текст программы

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

Листинг 6.1 – Текст файла prog6.cpp

|  |
| --- |
| /\*  Преобразовать заданную последовательность чисел c путем удаления из нее  всех чисел больше заданного значения a  \*/  #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 << "Введите значение: ";  int value;  cin >> value;  // Удаление элементов из исходного массива  for (int i = 0; i < arr\_current\_size;)  {  if (arr[i] > value)  {  for (int j = i; j < arr\_current\_size - 1; j++)  {  arr[j] = arr[j + 1];  }  arr\_current\_size--;  }  else i++;  }  cout << "Результирующий массив (числа больше " << value << " были удалены): ";  // Распечатка результ массива  for (int i = 0; i < arr\_current\_size; i++)  {  cout << arr[i] << " ";  }  cout << endl;  system("pause");  return 0;  } |

## 

## Тестирование программы

Результат работы программы приведён на рисунке 6.1.



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., C++ 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.