Департамент профессионального образования Томской области

Областное государственное бюджетное профессиональное образовательное учреждение среднего профессионального образования

**«ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

Специальность 09.02.07 «Информационные системы и программирование»

Отчёт о лабораторно-практической работе №9 по дисциплине  
«Основы алгоритмизации и программирования»

Разработка Функций

Студенты

«число» год                            Степанов М. Т.

\_\_\_\_\_\_\_\_\_\_\_\_\_ Чуприков Д. Э.

Преподаватель

«число» месяц год оценка                            Жабин Д. И.

Томск — 2022

**Оглавление**

[1 Постановка задачи 3](#_Toc81760271)

[2 Теоретические сведения 4](#_Toc81760272)

[3 Ход работы 5](#_Toc81760273)

[4 Результаты 6](#_Toc81760274)

[Приложение А. Исходный текст 7](#_Toc81760275)

[Приложение Б. Ответы на вопросы 9](#_Toc81760276)

# Постановка задачи

Цель работы: научиться писать программы с использованием структур на языке C++ в среде разработки Visual Studio 2022

Задачи:

1. Создать проект в Visual Studio.
2. Написать программы для их решения.
3. Выполнить программы.

Задание №1

Написать процедуру перемножения двух многочленов, заданных массивами коэффициентов, результатом которой является многочлен, заданный массивом коэффициентов.

Задание №2

Найти все простые натуральные числа, не превосходящие заданного числа n, двоичная запись которых представляет собой палиндром, т.е. читается одинаково слева направо и справа налево.

**Теоретические сведения**

# Функции — это блок кода, который вы можете использовать в любом участке вашей программы неограниченное количество раз.

Чтобы создать функцию, нужно использовать конструкцию, которая находится пониже:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <тип данных, который будет возвращаться функцией> <имя> (<аргументы функции>)  {    < блок      кода >  } |

**Тип данных функции**. В самом начале нам нужно указать тип данных, который в конечном итоге будет передавать функция.

Но если мы не собираемся ничего передавать, а например, хотим просто вывести строку, то на месте <тип данных> можно указать тип void.

**Имя функции.**Нам нужно задать функции имя (исключениями являются зарезервированные слова в C++, имена, начинающиеся с цифр, а также имена, разделенные пробелом).

Лучше всего задавать такое имя, которое будет говорить вам в будущем, за что отвечает эта функция. И тогда в будущем вам не придется  вспоминать, за что она отвечает.

**Аргументы функции.**В скобках (после имени функции) могут находиться аргументы функции. **Аргумент функции** — это значение, которое можно передать функции при ее вызове. Если аргумент функции не один, а их несколько, то их нужно разделять запятой.

b

a

j = 0

2

1

0

j = 2

j = 1

i = 2

i = 1

0

1

2

3

4

5

6

7

8

result

result[i \* b.size() + j] = a[i] \* b[j]

i = 0

2

3

4

2

1

3

# Ход работы

Создали проект в среде программирования Visual Studio 2022 «ОАиП ЛР - 9.1».

Пишем код первой программы (см. листинг А.1)

Результат работы программы:

|  |
| --- |
| Enter two polynomials:  Enter the number of terms: 2  term1 = 2  term2 = 3.5  Enter the number of terms: 3  term1 = 2.11  term2 = -3  term3 = 4.5  The product: 4.22, -6, 9, 7.385, -10.5, 15.75 |

Создали проект в среде программирования Visual Studio 2019 «ОАиП ЛР - 9.2».

Пишем код второй программы (см. листинг А.2)

Результат работы программы:

|  |
| --- |
| n = 5000  All prime natural numbers  that are palindromes in thier binary form  and less than 5000:  3 (11)  5 (101)  7 (111)  17 (10001)  31 (11111)  73 (1001001)  107 (1101011)  127 (1111111)  257 (100000001)  313 (100111001)  443 (110111011)  1193 (10010101001)  1453 (10110101101)  1571 (11000100011)  1619 (11001010011)  1787 (11011111011)  1831 (11100100111)  1879 (11101010111)  4889 (1001100011001) |

# Результаты

Мы научились писать функции на языке C++ в среде разработки Visual Studio 2022.

# Приложение А. Исходный текст

Листинг А.1

|  |
| --- |
| main.cpp:  #include <iostream>  #include <iomanip>  #include <vector>  #include <stdexcept>  void PrintPolynomyal(const std::vector<float>& polynomial)  {  const std::streamsize FLOAT\_PRECISION = 4;  if (polynomial.size() > 0)  {  std::cout << std::setprecision(FLOAT\_PRECISION) << polynomial[0];  for (int i = 1; i < polynomial.size(); ++i)  {  std::cout << ", " << std::setprecision(FLOAT\_PRECISION) << polynomial[i];  }  }  }  std::vector<float> ScanPolynomial()  {  std::cout << std::endl << "Enter the number of terms: ";  size\_t size = 0;  std::cin >> size;  std::vector<float> result(size);  for (int i = 0; i < size; ++i)  {  std::cout << "term" << (i + 1) << " = ";  std::cin >> result[i];  }  return result;  }  std::vector<float> MultiplyPolynomials  (const std::vector<float>& a, const std::vector<float>& b)  {  size\_t resultSize = a.size() \* b.size();  std::vector<float> result(resultSize);  for (int i = 0; i < a.size(); ++i)  {  for (int j = 0; j < b.size(); ++j)  {  result[i \* b.size() + j] = a[i] \* b[j];  }  }  return result;  }  int main()  {  try  {  std::cout << "Enter two polynomials:" << std::endl;  std::vector<float> a = ScanPolynomial();  std::vector<float> b = ScanPolynomial();  std::cout << std::endl << "The product: ";  PrintPolynomyal(MultiplyPolynomials(a, b));  }  catch (std::exception& e)  {  std::cout << std::endl << "Error: " << e.what();  return -1;  }  return 0;  } |
|  |

Листинг А.2

|  |
| --- |
| main.cpp:  #include <iostream>  #include <vector>  #include "Binary.h"  bool IsPrime(int number)  {  if (number < 1)  {  throw "The number must be whole";  }  if (number == 1)  {  return false;  }  for (int i = 2; i <= static\_cast<int>(number / 2); ++i)  {  if (number % i == 0)  {  return false;  }  }  return true;  }  int FoundNextPrime(int from)  {  while (true)  {  ++from;  if (IsPrime(from))  {  return from;  }  }  }  void FoundNumbers(int n)  {  std::cout << "All prime natural numbers"  << std::endl << "that are palindromes in thier binary form"  << std::endl << "and less than " << n << ":\n" << std::endl;  int num = 2;  while (num < n)  {  Binary binaryNum(num);  if (binaryNum.IsPalindrom())  {  std::cout << " " << num << " (" << binaryNum.ToString() << ")\n";  }  num = FoundNextPrime(num);  }  std::cout << std::endl;  }  int main()  {  std::cout << "n = ";  int n = 0;  std::cin >> n;  std::cout << std::endl;  FoundNumbers(n);  }  Binary.cpp:  #include "Binary.h"  #include <cmath>  std::vector<bool> Binary::BitsFromDecimal(int decimalValue)  {  size\_t n = static\_cast<size\_t>(ceil(log2(decimalValue + 1)));  std::vector<bool> result(n);  for (size\_t i = 0; i < n; ++i)  {  int a = static\_cast<int>(pow(2, n - i - 1));  if (decimalValue >= a)  {  result[i] = true;  decimalValue -= a;  if (decimalValue == 0)  {  break;  }  }  }  return result;  }  Binary::Binary(const std::vector<bool>& bits)  {  m\_bits = bits;  }  Binary::Binary(int decimalValue) : Binary(BitsFromDecimal(decimalValue)) {}  Binary::Binary() : Binary(0) {}  std::vector<bool> Binary::GetBits()  {  return m\_bits;  }  bool Binary::IsPalindrom()  {  size\_t length = static\_cast<size\_t>(m\_bits.size() / 2);  for (int i = 0; i < length; ++i)  {  if (m\_bits[i] != m\_bits[m\_bits.size() - i - 1])  {  return false;  }  }  return true;  }  std::string Binary::ToString()  {  const char ZERO\_CHAR = '0';  const char ONE\_CHAR = '1';  const std::string ZERO\_VALUE = "0";  size\_t size = m\_bits.size();  if (size == 0)  {  return ZERO\_VALUE;  }  std::string result(size, ZERO\_CHAR);  for (int i = 0; i < size; ++i)  {  if (m\_bits[i])  {  result[i] = ONE\_CHAR;  }  else  {  result[i] = ZERO\_CHAR;  }  }  return result;  }  Binary.h:  #pragma once  #include <vector>  #include <string>  class Binary  {  private:  std::vector<bool> m\_bits;  static std::vector<bool> BitsFromDecimal(int decimalValue);  public:  Binary(const std::vector<bool>& bits);  Binary(int decimalValue);  Binary();  std::vector<bool> GetBits();  bool IsPalindrom();  std::string ToString();  }; |
|  |

# Приложение Б. Ответы на вопросы

1. *Вопрос?*

Ответ.

1. *Вопрос?*

Ответ.

1. *Вопрос?*

Ответ.