Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы алгоритмизации и программирования»

ОТЧЕТ

по лабораторной работе №8

«Функции»

Вариант 1

Подготовил: Абакумов Г.Е.

Проверил: Усенко Ф.В.

Минск 2024

**Цель работы:** сформировать навыки и умения обработки структурированных типов данных, организованных в виде функций.

Задание 1: Дано натуральное число n. Выясните, можно ли представить данное число в виде произведения трех последовательных натуральных чисел.

Задание 2: Используя перегрузку методов, создайте программу:

а) для сложения целых чисел;

б) для сложения комплексных чисел.

#include <iostream>

#include <string>

using namespace std;

bool number\_representation(int n);

int addition(int a, int b);

string addition(string a, string b);

void converte(string x, string\* nums\_x);

int main(){

int input\_num;

cout << "Enter number:" << endl;

cin >> input\_num;

if (number\_representation(input\_num)) cout << "This number can be represented as the product of 3 consecutive natural numbers" << "\n\n";

if (!number\_representation(input\_num)) cout << "This number cannot be represented as the product of 3 consecutive natural numbers" << "\n\n";

cout << "Result: " << addition("-82i - 0.3", "-2 + 4i") << "\n\n";

cout << "Result: " << addition(4, 8) << "\n\n";

return 0;

}

bool number\_representation(int n) {

int num = 1;

while (true) {

int num\_multiply = num \* (num + 1) \* (num + 2);

if (num\_multiply < n) num += 1;

else if (num\_multiply == n) return true;

else return false;

}

}

int addition(int a, int b)

{

return a + b;

}

string addition(string a, string b) {

string nums\_a[2];

string nums\_b[2];

converte(a, nums\_a);

converte(b, nums\_b);

//imaginary part at index 0, real part at index 1

double numbers[2];

numbers[0] = stod(nums\_a[1]) + stod(nums\_b[1]);

numbers[1] = stod(nums\_a[0]) + stod(nums\_b[0]);

char answer[200];

if (numbers[1] < 0) snprintf(answer, sizeof answer, "%.1fi%.1f", numbers[0], numbers[1]);

if (numbers[1] >= 0) snprintf(answer, sizeof answer, "%.1fi+%.1f", numbers[0], numbers[1]);

return answer;

}

void converte(string x, string\* nums\_x) {

//removing spaces from equation string

for (int i = 0; i < 2; i++) {

if (x[i] == ' ') {

x.erase(i, 1);

}

}

//Splitting a string into real and imaginary parts

for (int i = 0; i < x.size(); i++) {

if (!(x[i] == '+' || x[i] == '-') || i == 0) {

nums\_x[0] += x[i];

}

else {

for (; i < x.size(); i++) {

nums\_x[1] += x[i];

}

break;

}

}

//checking that the imaginary part is at index 1

for (char l : nums\_x[0]) {

if (l == 'i') {

swap(nums\_x[0], nums\_x[1]);

break;

}

}

//Filling the empty rows of the array with zeros

for (int i = 0; i < 2; i++) {

if (!(nums\_x[i][0] == '-' || nums\_x[i][0] == '+' || isdigit(nums\_x[i][0]))) nums\_x[i] += "0";

}

//Removing all characters except numbers

for (int i = 0; i < 2; i++) {

for (int j = 0; j < nums\_x[i].size(); j++) {

if (!(nums\_x[i][j] == '-' || nums\_x[i][j] == '+' || nums\_x[i][j] == '.' || isdigit(nums\_x[i][j]))) {

nums\_x[i].erase(j, 1);

}

}

}

}

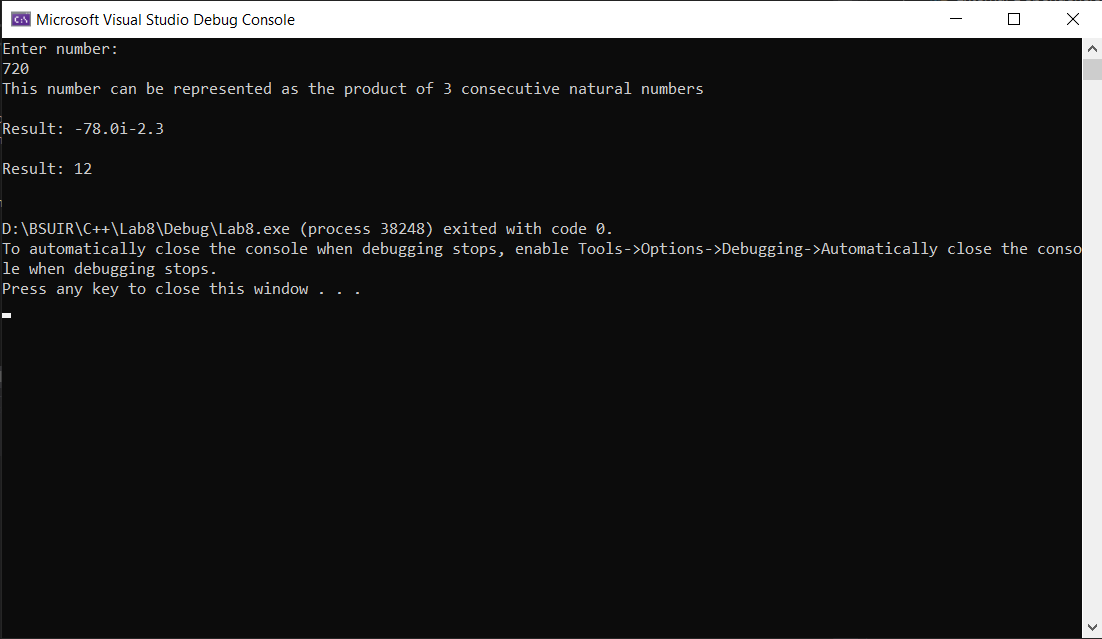
Результат работы программы представлен на рисунке 1.

Рисунок 1 – Результат работы программы

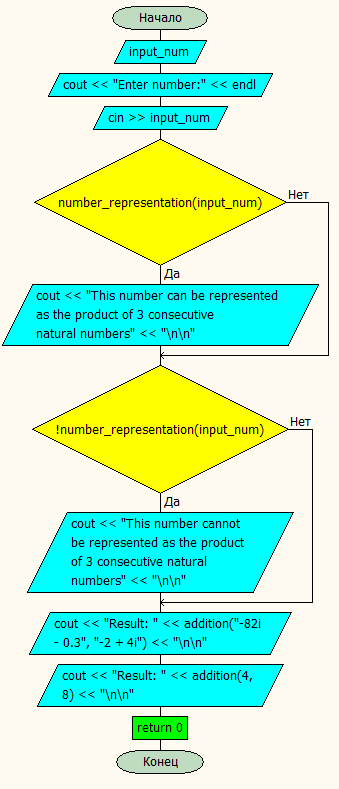
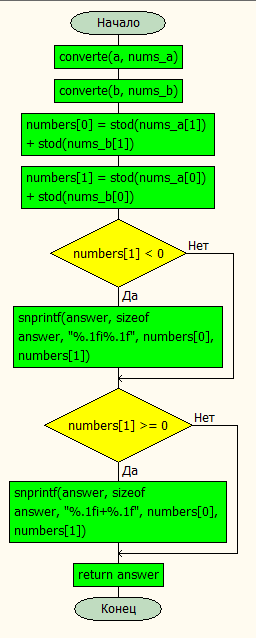
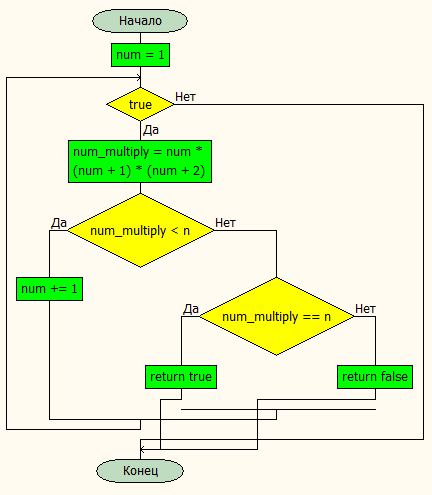
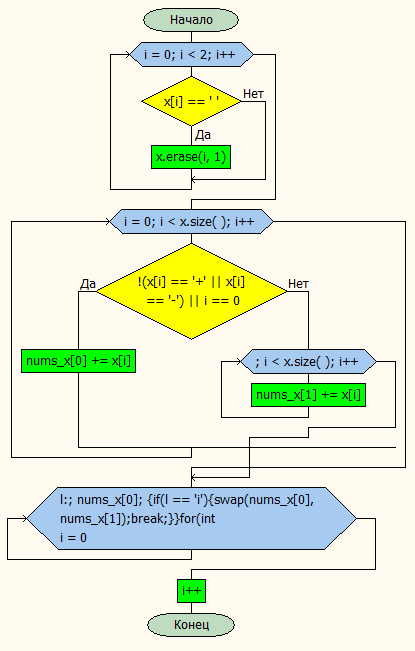
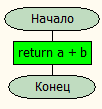
Блок-схема работы программы представлена на рисунке 2.

Рисунок 2

**Вывод:** в ходе выполнения работы была достигнута цель данной лабораторной работы: сформировать навыки и умения обработки структурированных типов данных, организованных в виде функций.