Федеральное государственное автономное образовательное учреждение высшего образования

«ОМСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Информатика и вычислительная техника.

**Лабораторная работа №8**

по дисциплине «Программирование» на тему:

**«Программирование с применением функций.»**

Выполнил: студент группы ИВТ-244 Шмидт Антон Владиславович

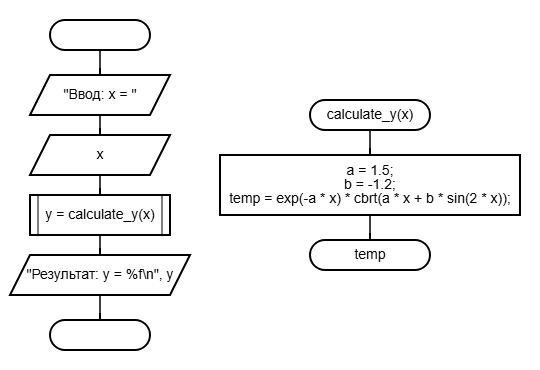
Проверил: ассистент кафедры ИВТ Горшенин Алексей Юрьевич

Омск 2024

***Задача 1*** (программа 8\_1)

Задача: написать и отладить программу для задачи из лабораторной работы 1 (программа1\_1) с применением подпрограммы-функции с параметрами. Вычисление функции (или обработку массива) выполнить в подпрограмме, а ввод исходных данных и вывод результатов в основной программе.

Схема алгоритма:



Решение кодом:

#include <locale.h>

#include <stdio.h>

#include <math.h>

#define a 1.5

#define b -1.2

double calculate\_y(double x) {

return exp(-a \* x) \* cbrt(a \* x + b \* sin(2 \* x));

}

void main8\_1() {

setlocale(LC\_ALL, "ru\_RU");

double x, y;

printf("Ввод: x = ");

scanf\_s("%lf", &x);

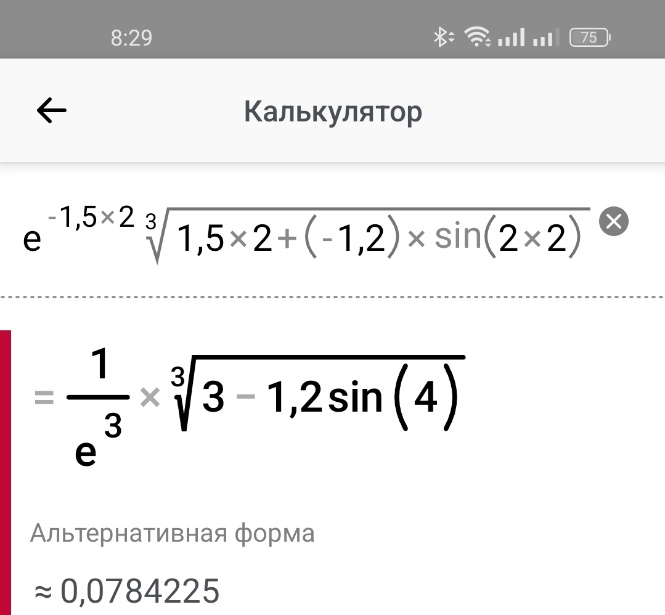
y = calculate\_y(x);

printf("Результат: y = %f\n", y);

}

Результат работы:

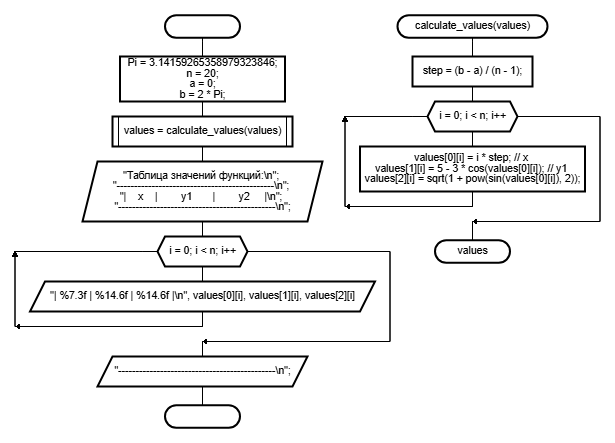




***Задача 2*** (программа 8\_2)

Задача: написать и отладить программу для задачи из лабораторной работы 6 (программа6\_2 для двух массивов) с применением подпрограммы-функции с параметрами. Вычисление функции (или обработку массива) выполнить в подпрограмме, а ввод исходных данных и вывод результатов - в основной программе.

Схема алгоритма:



Решение кодом:

#include <locale.h>

#include <stdio.h>

#include <math.h>

#define Pi 3.14159265358979323846

#define n 20

#define a 0

#define b 2 \* Pi

void calculate\_values(double values[3][n]) {

double step = (b - a) / (n - 1);

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

values[0][i] = i \* step;

values[1][i] = 5 - 3 \* cos(values[0][i]);

values[2][i] = sqrt(1 + pow(sin(values[0][i]), 2));

}

}

void main8\_2() {

setlocale(LC\_ALL, "ru\_RU");

double values[3][n];

calculate\_values(values);

printf("Таблица значений функций:\n");

printf("---------------------------------------------\n");

printf("| x | y1 | y2 |\n");

printf("---------------------------------------------\n");

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

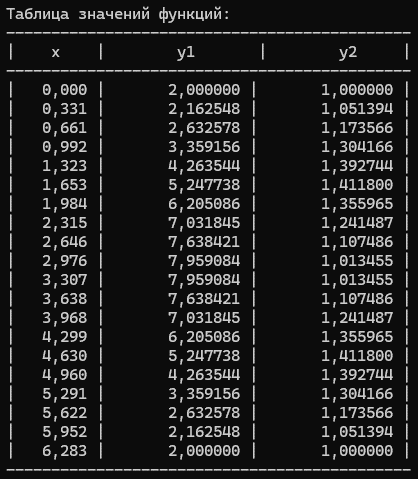
printf("| %7.3f | %14.6f | %14.6f |\n", values[0][i], values[1][i], values[2][i]);

}

printf("---------------------------------------------\n");

}

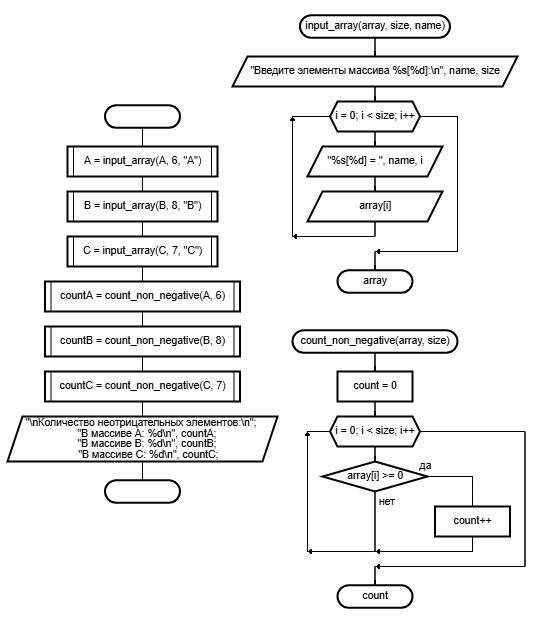
Результат работы:



***Задача 3*** (программа 8\_3)

Задача: даны три одномерных массива целых чисел A[6], B[8] и С[7]. Подсчитать количество неотрицательных элементов в каждом.

Схема алгоритма:



Решение кодом:

#include <locale.h>

#include <stdio.h>

void input\_array(int array[], int size, const char\* name) {

printf("Введите элементы массива %s[%d]:\n", name, size);

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

printf("%s[%d] = ", name, i);

scanf\_s("%d", &array[i]);

}

}

int count\_non\_negative(const int array[], int size) {

int count = 0;

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

if (array[i] >= 0) {

count++;

}

}

return count;

}

void main8\_3() {

setlocale(LC\_ALL, "ru\_RU");

int A[6], B[8], C[7];

input\_array(A, 6, "A");

input\_array(B, 8, "B");

input\_array(C, 7, "C");

int countA = count\_non\_negative(A, 6);

int countB = count\_non\_negative(B, 8);

int countC = count\_non\_negative(C, 7);

printf("\nКоличество неотрицательных элементов:\n");

printf("В массиве A: %d\n", countA);

printf("В массиве B: %d\n", countB);

printf("В массиве C: %d\n", countC);

}

Результат работы:

