МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Государственное образовательное учреждение

высшего профессионального образования

«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ

ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»

ОТЧЁТ ПО ВЫПОЛНЕНИЮ КОНТРОЛЬНОЙ РАБОТЫ

ПО КУРСУ «ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ»

Вариант 5

Выполнил: студент гр. ЗПИт-??-??

Иванов Рустем Карлович

Преподаватель: к.ф.-м.н., доц.

Петрова Н.К.

Казань 201?

**Задание 1**

Написать функцию AddRightDigit (D, K), добавляющую к целому положительному числу К справа цифру D. Для D организовать проверку на принадлежность диапазону [0;9]. C помощью этой функции последовательно добавить к числу К справа заданные цифры D1 и D2.

**Текст программы:**

#include <iostream>

#include <string>

using namespace std;

void AddRightDigit(char D, int K) {

if (D < '0' || D > '9') {

cout << "Nevernoe D" << endl;

return;

}

string s = to\_string(K);

s += D;

cout << "otvet: " << s << endl;

}

int main() {

int K;

cout << "Vvedite K: ";

cin >> K;

char D1, D2;

cout << "Vvedite D1: ";

cin >> D1;

cout << "Vvedite D2: ";

cin >> D2;

AddRightDigit(D1, K);

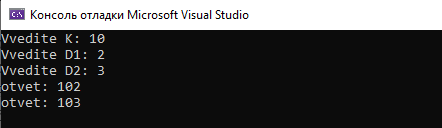
AddRightDigit(D2, K);

return 0;

}

**Решение:**

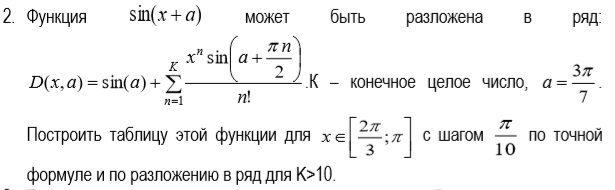
*Реакция программы на ввод однозначного числа N*



**Описание программы и полученного решения**

Функция AddRightDigit содержит 2 параметра. D – символ, функция проверяет, является ли D цифрой, если нет, то функция завершается. K – число, которое вводится пользователем и передаётся в функцию. Далее функция конвертирует число K в тип string, добавляет справа цифру и выводит результат.

**Задание 2**



**Текст программы:**

// Подключаем библиотеки

#include <iostream>

#include <iomanip>

#include <math.h>

#define M\_PI 3.14159265358979323846

// Используем стандартное пространство имён

using namespace std;

int main() {

double a, b, h, c;

c = 3 \* M\_PI / 7;

a = 2 \* M\_PI / 3;

b = M\_PI;

h = M\_PI / 10;

int K;

cout << "Vvedite K" << endl;

cin >> K;

if (K <= 10) {

cout << "Nevernoe K" << endl;

return 0;

}

// Инициализируем переменные

double x, y, s, p, d;

// Вводим данные согласно заданию

x = a;

cout << setw(15) << "x" << setw(15) << "y" << setw(15) << "s" << endl;

// Пробегаемся по всем x от a до б с шагом h

do {

// Первый член ряда

s = sin(a);

p = 1;

// Остальные члены ряда

for (int i = 1; i <= K; i++) {

d = sin(a + M\_PI \* i / 2);

p \*= x/i;

if (isnan(p)) break; // Если p - не число (nan), то выходим из цикла

s += p \* d;

}

// Сама функция

y = sin(x + a);

// Вывод

cout << setw(15) << x << setw(15) << y << setw(15) << s << endl;

// Делаем шаг

x += h;

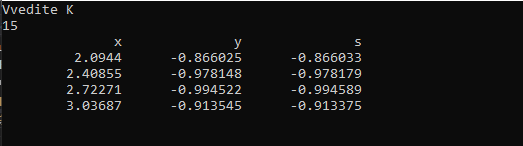
} while (x <= b + h / 2);

cout << endl;

return 0;

}

**Решение:**



**Описание программы и полученного решения**

Код программы соответствует поставленным условиям: для вычисления суммы ряда найдены рекуррентные соотношения. x – отражает переменную ряда, которая меняется с шагом h, y – функция, посчитанная напрямую, с помощью библиотеки math.h, s – результат вычисления ряда. Результаты на консоли оформлены в понятном виде. Исходные данные введены в формате с плавающей точкой, как того требует задание.

**Задание 3**

Построить массив из разностей между точным и приближенным значением функции определить в каких пределах изменяется эта разность.

***Текст программы:***

// Подключаем библиотеки

#include <iostream>

#include <iomanip>

#include <math.h>

#include <vector>

#include <algorithm>

#define M\_PI 3.14159265358979323846

// Используем стандартное пространство имён

using namespace std;

int main() {

vector <double> raznost;

double a, b, h, c;

c = 3 \* M\_PI / 7;

a = 2 \* M\_PI / 3;

b = M\_PI;

h = M\_PI / 10;

int K;

cout << "Vvedite K" << endl;

cin >> K;

if (K <= 10) {

cout << "Nevernoe K" << endl;

return 0;

}

// Инициализируем переменные

double x, y, s, p, d;

// Вводим данные согласно заданию

x = a;

cout << setw(15) << "x" << setw(15) << "y" << setw(15) << "s" << endl;

// Пробегаемся по всем x от a до б с шагом h

do {

// Первый член ряда

s = sin(a);

p = 1;

// Остальные члены ряда

for (int i = 1; i <= K; i++) {

d = sin(a + M\_PI \* i / 2);

p \*= x/i;

if (isnan(p)) break; // Если p - не число (nan), то выходим из цикла

s += p \* d;

}

// Сама функция

y = sin(x + a);

// Вывод

cout << setw(15) << x << setw(15) << y << setw(15) << s << endl;

raznost.push\_back(abs(s - y));

// Делаем шаг

x += h;

} while (x <= b + h / 2);

double maxElement = \*max\_element(raznost.begin(), raznost.end());

double minElement = \*min\_element(raznost.begin(), raznost.end());

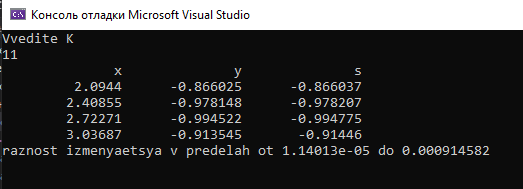
cout << "raznost izmenyaetsya v predelah ot " << minElement << " do " << maxElement << endl;

cout << endl;

return 0;

}

**Решение:**



**Описание полученного решения**

Для решения задания был создан вектор, в который добавлялась на каждой итерации разность между вычисленным значением функции и ряда.

Затем с помощью библиотеки algorithm были найдены максимальный и минимальный элементы вектора и выведены на экран.

Отчёт сдан 15 января 2019 года

/Подпись/