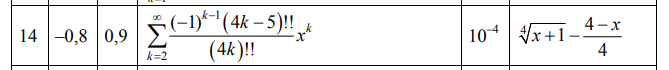
**Шараев Владислав Николаевич 020602**

**Лабораторная работа №8**

**Вариант 14**

***Условие:*** Вывести на экран таблицу значений функции Y(x) и ее разложения в ряд S(x) с точностью ε (табл. 8.1). Вывести число итераций, необходимое для достижения заданной точности. Вычисление S(x) и Y(x) оформить в виде функций. 

***Текст программы:***

#include <iostream>

#include <cstdio>

#include <cmath>

using namespace std;

const double EPS = 1e-4; // Константа для указания точности

void DeltaCalculating(double &delta, int k) { // Вычисление рекуррента

for (int p = min(2, 4 \* (k - 1)); p <= 4 \* k; p += 2) {

delta /= p;

}

for (int p = min(3, abs(4 \* (k - 1) - 5)); p <= 4 \* k - 5; p += 2) {

delta \*= p;

}

}

double Y(double x) { // Вычисление Y

return sqrt(sqrt(x + 1)) - (4 - x) / 4;

}

void Solve(double a, double b, double h) { // Основная функция для решения задачи

double delta = 1, sum;

int k;

for (double x = a; x < b + h / 2; x += h) { // Цикл с шагом h

k = 2; // Инициализация переменных

delta = x \* x;

sum = 0;

while (fabs(delta) > EPS) { // Цикл while для достижения точности EPS

delta \*= -1;

DeltaCalculating(delta, k); // Вызов функции для подсчета рекуррента

sum += delta; // Вычисление суммы

k++;

}

printf("Значение X = %.2lf\nКоличество итераций = %d\nЗначение рекуррента = %lf\nСумма = %lf\nЗначение Y = %lf\n\n", x, k - 2, delta, sum, Y(x)); // Вывод ответа для данного шага

}

}

int main()

{

setlocale(LC\_ALL, "Russian"); // Добавление поддержки кириллицы в консоли

Solve(-0.8, 0.9, 0.1); // Вызов основной функции для решения задачи

return 0;

}

***Результат работы программы:***

