**Курсовой проект №3**

Студент группы М8О-101Б-22 Анциборко Л. В., № по списку: 1.

Контакты: anciborko04@mail.ru

Преподаватель: Cахарин Н. А., каф. 806.

Входной контроль знаний с оценкой:

Итоговая оценка:

Подпись преподавателя:

Оглавление

[*Задание* 2](#_Toc124267281)

[*Вариант* 2](#_Toc124267282)

[*Алгоритм* 2](#_Toc124267283)

[*Код* 2](#_Toc124267284)

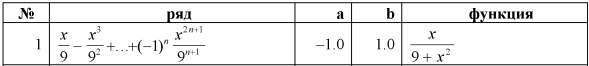
[*Примеры работы программы* 3](#_Toc124267285)

[*Вывод* 4](#_Toc124267286)

# *Задание*

Составить программу на Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования.

# *Вариант*



# *Алгоритм*

Программа получает на вход число *n*. Это значит, что на отрезке будет найдено *n – 1* равных отрезков, а программа будет работать с их концами.

В переменную result программа сохраняет значение функции в точке, посчитанное с помощью встроенных функций языка программирования. К переменной *sum* программа прибавляет слагаемые по формуле Тейлора.

# *Код*

#include <stdio.h>

#include <math.h>

#include <conio.h>

#define MAX\_NUM\_OF\_ITER 100

const long double k = 1000;

double find\_machine\_epsilon();

long double func(long double x);

long double taylor(double x, int n);

void main() {

    double abs\_eps = find\_machine\_epsilon();

    double rel\_eps = sqrt(abs\_eps);

    long double a = -1.0, b = 1.0, x = 0.0, result;

    int n, t;

    scanf("%d", &n);

    printf("epsilon %.16e\n", abs\_eps);

    long double step = (b - a) / n;

    printf("x \t\t sum of row \t\t\t function \t\t\t iter\t \n");

    for (long double x = a; x <= b; x += step) {

      for (n = 0; n < MAX\_NUM\_OF\_ITER; n++) {

        result = taylor(x, n);

        if (fabs(result) <= fmax(rel\_eps \* fabs(result), abs\_eps)) {

          break;

        }

      }

      printf("%.5Lf \t %.20Lf \t %.20Lf \t %d\t \n", x, result, func(x), n);

      result = 0.0;

    }

    getch();

}

double find\_machine\_epsilon() {

    long double eps = 1.0;

    while (1 + eps / 2.0 != 1)

        eps /= 2.0;

    return eps;

}

long double func(long double x) {

    return x / (9 + x \* x);

}

long double taylor(double x, int n) {

    long double sum = 0;

    for (int i = 0; i <= n; i++) {

      sum += (pow(-1, i) \* pow(x, 2 \* i + 1) / pow(9, i + 1));

    }

    return sum;

}

# *Примеры работы программы*

6

epsilon 1.0842021724855044e-19

x sum of row function iter

-1.00000 -0.09999999999999999441 -0.10000000000000000000 100

-0.66667 -0.07058823529411764388 -0.07058823529411764705 100

-0.33333 -0.03658536585365853465 -0.03658536585365853658 100

0.00000 0.00000000000000000001 0.00000000000000000001 0

0.33333 0.03658536585365853465 0.03658536585365853659 100

0.66667 0.07058823529411764388 0.07058823529411764706 100

12

epsilon 1.0842021724855044e-19

x sum of row function iter

-1.00000 -0.09999999999999999441 -0.10000000000000000000 100

-0.83333 -0.08595988538681949139 -0.08595988538681948424 100

-0.66667 -0.07058823529411764388 -0.07058823529411764705 100

-0.50000 -0.05405405405405405105 -0.05405405405405405405 100

-0.33333 -0.03658536585365853465 -0.03658536585365853658 100

-0.16667 -0.01846153846153846052 -0.01846153846153846153 100

0.00000 0.00000000000000000001 0.00000000000000000001 0

0.16667 0.01846153846153846052 0.01846153846153846155 100

0.33333 0.03658536585365853465 0.03658536585365853659 100

0.50000 0.05405405405405405105 0.05405405405405405407 100

0.66667 0.07058823529411764388 0.07058823529411764707 100

0.83333 0.08595988538681949139 0.08595988538681948425 100

18

epsilon 1.0842021724855044e-19

x sum of row function iter

-1.00000 -0.09999999999999999441 -0.10000000000000000000 100

-0.88889 -0.09079445145018915178 -0.09079445145018915510 100

-0.77778 -0.08097686375321336223 -0.08097686375321336761 100

-0.66667 -0.07058823529411764388 -0.07058823529411764705 100

-0.55556 -0.05968169761273209868 -0.05968169761273209548 100

-0.44444 -0.04832214765100670895 -0.04832214765100671140 100

-0.33333 -0.03658536585365853465 -0.03658536585365853658 100

-0.22222 -0.02455661664392905733 -0.02455661664392905866 100

-0.11111 -0.01232876712328767055 -0.01232876712328767122 100

0.00000 0.00000000000000000001 0.00000000000000000001 0

0.11111 0.01232876712328767055 0.01232876712328767124 100

0.22222 0.02455661664392905733 0.02455661664392905867 100

0.33333 0.03658536585365853465 0.03658536585365853659 100

0.44444 0.04832214765100670895 0.04832214765100671141 100

0.55556 0.05968169761273209868 0.05968169761273209550 100

0.66667 0.07058823529411764388 0.07058823529411764706 100

0.77778 0.08097686375321336223 0.08097686375321336761 100

0.88889 0.09079445145018915178 0.09079445145018915512 100

# *Вывод*

Мной было освоено составление программ на Си, которые вычисляют значение элементарных функций двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования.