**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Факультет №8 «Информационные технологии и прикладная математика»**

**Кафедра 805 «Математическая кибернетика»**

Курсовая работа

по курсу «Вычислительные системы»

1 семестр

Задание 1

Схема домашнего компьютера

**Автор работы:**

студент 1 курса, группы М8О-103Б-21

Березнев Н.В.

**Руководитель проекта:**

Севастьянов В.С.

**Дата сдачи:**

Москва, 2021

**Цель работы:**

Составить программу на Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования. В качестве аргументов таблицы взять точки разбиения отрезка [a, b] на n равных частей (n + 1 точка, включая концы отрезка), находящихся в рекомендованной области хорошей точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью ε × k, где ε – машинное эпсилон аппаратно реализованного вещественного типа для данной ЭВМ, а k – экспериментально подбираемый коэффициент, обеспечивающий приемлемую сходимость. Число итераций должно ограничиваться сверху числом порядка 100. Программа должна сама определить машинное ε и обеспечивать корректные размеры генерируемой таблицы.

**Задание варианта № 4:**



**Алгоритм:**

- Вычисляем машинное эпсилон через разность дробных чисел.

- После ввода пользователем числа разбиений отрезка вычисляем количество точек разбиения отрезка и вычисляем интервал между ними.

- Запускаем цикл с числом проходов, равным количеству точек разбиений.

- В цикле с помощью ряда Тейлора высчитываем значение функции до тех пор, пока значение элемента ряда Тейлора не будет превышать машинное эпсилон.

- Высчитываем значение функции через библиотеку math.h.

- Выводим в формате таблицы номер прохода, значения функции, вычисленные через ряд Тейлора и через функции библиотеки math.h, а также число повторов цикла в ряде Тейлора.

**Описание программы:**

Использованные переменные:

- double eps – Машинное эпсилон для данного компьютера.

- double lgf – Значение функции.

- double lg – Значение функции ln(2).

- double func – Значение фрагмента ряда Тейлора для конкретного n.

- double sum – Значение всего ряда Тейлора.

- double step – Шаг изменения x.

- double X – Значение переменной, используемой в качестве аргумента в функции и ряде Тейлора.

- double a – Левая граница отрезка, из которого берется X.

- double b – Правая граница отрезка, из которого берется X.

- int n – Параметр, используемый в ряде Тейлора.

- int I – Счётчик цикла.

- int iter – Счётчик числа итераций в ряде Тейлора.

- int part – Число разбиений отрезка.

Входные данные:

- Целое число разбиений отрезков

Выходные данные:

- Значение машинного эпсилон

- Таблица с вычисленными двумя способами - по формуле Тейлора и при помощи встроенных функций библиотеки math.h - значениями функции в точках разбиения заданного отрезка и на его концах.

**Код программы:**

#include <stdio.h>

#include <math.h>

int Line(void)

{

int i;

printf("|");

for (i = 0; i < 137; i++) {

printf("-");

}

printf("|\n");

return 0;

}

int main(void)

{

double eps;

double lgf;

double lg;

double func;

double sum;

double step;

double X;

double a;

double b;

int n;

int i;

int iter;

int part;

a = -1.0;

b = 1.0;

iter = 0;

X = a;

eps = (7.0 / 3.0 - 4.0 / 3.0 - 1.0);

printf("Машинное эпсилон: %.42f\n", eps);

lg = log(2);

printf("Введите число разбиений отрезка [-1, 1]: ");

scanf("%d", &part);

step = (b - a) / part;

Line();

printf("| Номер шага | Значение X: | Значение функции по формуле Тейлора: | Значение функции по встроенной функции ЯП: | Число итераций |\n");

Line();

for (i = 0; i <= part; i++) {

n = 1;

func = 1;

sum = 0;

while ((func > eps || func < -eps) && (n < 100)) {

func = pow((-1), n - 1) \* (pow(X, n) / (n \* pow(2, n)));

sum = sum + func;

n = n + 1;

iter = iter + 1;

}

sum = sum + lg;

lgf = log(2 + X);

printf("| %2d\t | %.2f\t | %.42f | %.42f | %2d |\n", i, X, sum, lgf, iter);

X = X + step;

iter = 0;

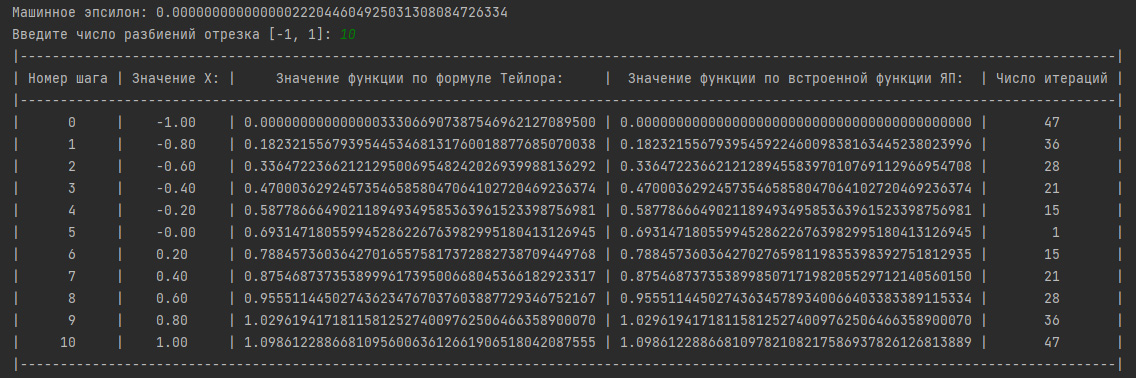
}

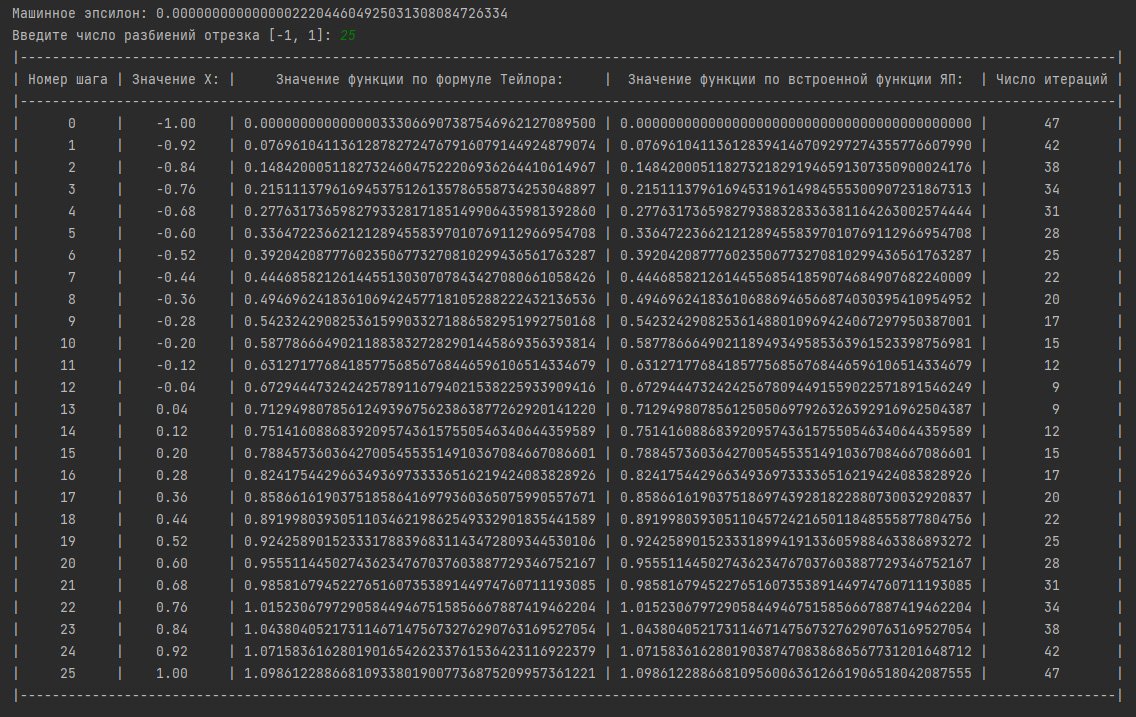
Line();

return 0;

}

**Результат:**





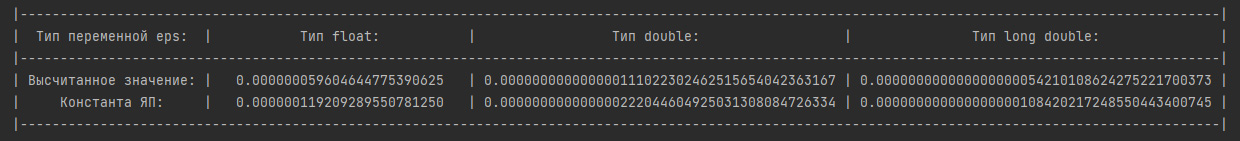
**Заключение:**

Исходя из данных таблицы, можно сделать вывод, что значения ряда Тейлора имеют отличия от встроенной функции после 15 знака после запятой. Это означает, что, несмотря на точность данного метода задания функций, он не является совершенным

Формула Тейлора сводит вычисление трансцендентных функций к алгебраическим полиномам. Однако этот простой способ не применяется на практике ввиду большой ресурсоёмкости и значительной погрешности.

**Дополнительное задание:**

Значение машинного эпсилон в Си:

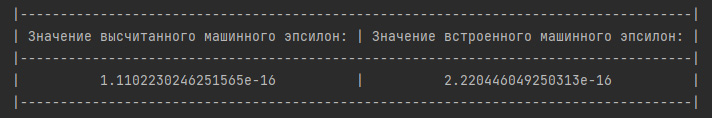


Код программы:

#include <stdio.h>  
#include <float.h>  
  
int Line(void)

{  
 int i;  
 printf("|");  
 for (i = 0; i < 150; i++) {  
 printf("-");  
 }  
 printf("|\n");  
}  
  
int main(void)  
{  
 float eps\_f;  
 double eps\_d;  
 long double eps\_l;  
  
 eps\_f = 1;  
 eps\_d = 1;  
 eps\_l = 1;  
  
 while (1 + eps\_f > 1) {  
 eps\_f = eps\_f / 2;  
 }  
 while (1 + eps\_d > 1) {  
 eps\_d = eps\_d / 2;  
 }  
 while (1 + eps\_l > 1) {  
 eps\_l = eps\_l / 2;  
 }  
 Line();  
 printf("| Тип переменной eps: |%24s %10s| %30s %17s| %33s %14s|\n", "Тип float:", "", "Тип double:", "", "Тип long double:", "");  
 Line();  
 printf("%9s %.24f | %.42f | %.42Lf |\n", "| Высчитанное значение: |", eps\_f, eps\_d, eps\_l);  
 printf("%9s %.24f | %.42f | %.42Lf |\n", "| Константа ЯП: |", FLT\_EPSILON, DBL\_EPSILON, LDBL\_EPSILON);  
 Line();  
}

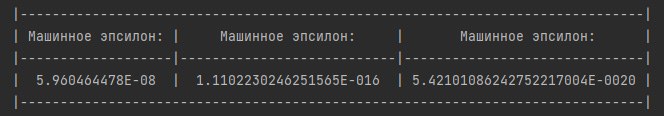
Значение машинного эпсилон в python:



Код программы:

import sys  
eps = 1.0  
while 1.0 + eps > 1.0:  
 eps = eps / 2  
def line():  
 print("|", end = '')  
 for i in range(0, 84):  
 print('-', end = '')  
 print("|")  
line()  
print('| Значение высчитанного машинного эпсилон: | Значение встроенного машинного эпсилон: |')  
line()  
print('| ', float(eps), ' | ', sys.float\_info.epsilon,' |')  
line()

Значение машинного эпсилон в Pascal:



Код программы:

program Epsilon;

var

eps\_1: real;

eps\_2: single;

eps\_3: double;

eps\_4: extended;

begin

eps\_1 := 1;

eps\_2 := 1;

eps\_3 := 1;

eps\_4 := 1;

while eps\_1 + 1 > 1 do

eps\_1 := eps\_1 / 2;

while eps\_2 + 1 > 1 do

eps\_2 := eps\_2 / 2;

while eps\_3 + 1 > 1 do

eps\_3 := eps\_3 / 2;

while eps\_4 + 1 > 1 do

eps\_4 := eps\_4 / 2;

Writeln('|------------------------------------------------------------------------------|');

Writeln('| Машинное эпсилон: | Мaшинное эпсилон: | Машинное эпсилон: |');

Writeln('|-------------------|---------------------------|------------------------------|');

Writeln('| ', eps\_2, ' |', ' ', eps\_3, ' |', eps\_4, ' |');

Writeln('|------------------------------------------------------------------------------|');

end.