**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное** **учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»**

Факультет компьютерных наук

Департамент программной инженерии

# Отчет к домашнему заданию По дисциплине

**«Архитектура вычислительных систем»**

Работу выполнил:

Студент группы БПИ-194 Остапенко С.В.

**Москва 2020**

**Задание**

Разработать программу, вычисляющую с помощью степенного ряда с точностью не хуже 0,05% значение биномиальной функции для заданного параметра m и x (использовать FPU)

**Решение**

Функция может быть разложена в ряд Тейлора следующим образом:

Получим общий член ряда разделив выражение под знаком суммы для итераций на выражение для итераций:

Чтобы проверить правильность степенного ряда следует рассчитать точное значение функции при переданных параметрах . Так как в наборе команд FPU отсутствует команда, возводящая число в некоторую степень, то нам нужно циклом умножать само на себя раз

Для решения задачи нам также понадобится цикл вычисляющий сумму членов ряда Тейлора, заканчивающийся в тот момент, когда модуль очередного слагаемого будет меньше заданной точности, а именно 0.05% (0.0005)

Для ввода исходных данных будет использоваться функция стандартной библиотеки C: char \*gets(char \*str), считывающая строку и помещающая ее в массив символов на который указывает указатель str.

Для преобразования полученной строки в вещественное число применим функцию sscanf, которая распознает и считывает данные по шаблону строки

Для вывода данных в консоль воспользуемся функцией printf, которая форматирует данные и выводит их по заданному шаблону.

Для завершения работы программы будет использоваться функция ExitProcess

Программа была разбита на следующие функции:

* double teylorRow(int m, double x, double eps) – функция для вычисляющая значение с точностью eps.

В данной функции используется две локальные переменные:

* + int t – используются для вычисления очередного слагаемого ряда
  + double a – хранит в себе значение очередного слагаемого ряда
* double function(double x, int m) – вычисляет точное значение

В данной функции используется одна локальная переменная:

* int t – используется для возведения в степень

**Текст программы**

format PE Console

entry start

include 'win32a.inc'

;

; Студент: Степан Остапенко

; Группа: БПИ194

; Вариант: 15

;

; Задача:

; Разработать программу, вычисляющую с

; помощью степенного ряда с точностью не

; хуже 0,05% значение биноминальной функции (1+x)^m

; для заданного параметра m и x (использовать FPU)

;

section '.data' data readable writeable

x1 dq ? ; Введённое пользователем значение x:

m dd ? ; Введённое пользователем значение m:

eps1 dd 0.0005 ; Точность 0.05%

msg1 db 'Enter x: ',0 ; Сообщения для ввода x

msg2 db 'Enter m: ',0 ; Сообщение для ввода m

msg3 db 'Wrong number.',13,10,0 ; Сообщения об ошибке

inDouble db '%lf',0 ; Ввод вещественного числа

inInt db '%d',0 ; Ввод целого числа

msg4 db 'Teylor row = %lg',13,10,0 ; Строка для вывода значения ряда тейлора

msg5 db '(1+x)^m = %lg',13,10,0 ; Строка для вывода значния функции

buf db 256 dup(0) ; Для парса вещественного числа

section '.code' code readable executable

start:

ccall [printf],msg1 ; Выводим в консоль Enter x:

ccall [gets],buf ; Считываем введенную строку

ccall [sscanf],buf,inDouble,x1 ; Парсим введенную строку

; Проверяем удалось ли преобразование

cmp eax,1

jz nextNum

; Иначе выводим сообщения об ошибке и считываем число еще раз

ccall [printf],msg3

jmp start

nextNum:

ccall [printf],msg2 ; Вывоодим в консоль Enter m:

push m ; Передаем в стек m

push inInt ; Передаем в стек строчку для чтения Int

call [scanf] ; Считываем m

add esp, 8 ; Отчищаем стек от переданных параметров

; Проверяем корректность введеного числа

mov ebx, [m]

cmp ebx, 0

jg m1

; Иначе выводим сообщения об ошибке и считываем число еще раз

ccall [printf],msg3

jmp ex

m1:

fld [eps1] ; Точность вычисления

sub esp, 8 ; Выделяем в стеке место под double

fstp qword [esp] ; Записать в стек double число

fld qword [x1] ; Введенное значение

sub esp, 8 ; Выделить в стеке место под double

fstp qword [esp] ; Записать в стек double число

push [m] ; Записываем в стек значение m

call teylorRow ; Вычислить (1+x)^m через ряд тейлора

add esp, 20 ; Удалить переданные параметры

sub esp, 8 ; Передать сумму ряда

fstp qword [esp] ; Функции через стек

push msg4 ; Формат сообщения

call [printf] ; Сформировать результат

add esp, 12 ; Коррекция стека

fld qword [x1] ; Введенное значение

sub esp, 8 ; Выделить в стеке место под double

fstp qword [esp] ; Записать в стек double число

push [m] ; Запись m в стек

call function ; Вычислить (1+x)^m

add esp, 16 ; Удалить переданные параметры

sub esp, 8 ; Передать точное значение (1+x)^m

fstp qword [esp] ; Функции через стек

push msg5 ; Формат сообщения

call [printf] ; Сформировать результат

add esp, 12 ; Коррекция стека

ex:

ccall [\_getch] ; Ожидание нажатия любой клавиши

push 0

call [ExitProcess]

; Описание:

; Вычисляет значение (1+x)^m с точностью eps

; Аргументы:

; int m - значение m

; double x - значение x

; double esp - точночть вычислений

; Вывод:

; Значение (1+x)^m вычисленное через ряд тейлора

;--------------------teylorRow(int m, double x, double eps)------------------------------

teylorRow:

push ebp

mov ebp,esp

sub esp,0ch ; Выделение места в стеке для локальных переменных

;Локальные переменные:

t equ ebp-0ch

a equ ebp-8h

;Переданные функции параметры:

m equ ebp+08h

x equ ebp+0ch

eps equ ebp+14h

;Вычисленное значение

fld qword [x] ; Загрузить х

fimul dword [m] ; x\*m

fstp qword [a] ; a = x\*m

fld1 ; 1

fldz ; s=0

mov ecx,1 ; n=1

m11:

fadd qword [a] ; s += a;

mov edx, [m] ; edx = m

inc ecx ; n++;

lea eax, [ecx-1] ; n-1

sub edx, eax ; m-n+1

cmp edx, 0 ; Проверяем что m-n+1 == 0

je m11 ; Пропускаем итерацию цикла

fld qword [a] ; a

fmul qword [x] ; a\*x

mov [t], edx ; t=m-n+1

fidiv dword [t] ; a\*x/(m-n+1)

lea eax,[ecx+1] ; n+1

mov [t],eax ; t=n+1

fidiv dword [t] ; a\*x/(2n-1)/(n+1)

fst qword [a] ; a = a\*x/(2n-1)/(n+1);

fabs ; |a|

fcomp qword [eps] ; сравнить |a| c eps

fstsw ax; ; перенести флаги сравнения в ах

sahf; ; занести ah в флаги процессора

jnb m11; ; Если |a|>=esp, продолжить цикл

faddp st1,st ; 1+полученная сумма

leave

ret

;----------------------------------------------------------------------------------------

; Описание:

; Вычисляет точное значение (1+x)^m

; Аргументы:

; double x - значение x

; int m - значение m

; Вывод:

; Точное значение функции

; Соглашение:

; Соглашение вызова через cdecl

; --------------------function(double x, int m)------------------------------------------

function:

push ebp

mov ebp,esp

sub esp,04h ; Выделение места в стеке для локальной переменной

;Локальные переменные:

t equ ebp-04h

;Вычисление значения

fld1 ; Для умножения

fld qword [ebp+12] ; x

fld1 ; 1

faddp st1, st ; x+1

mov ecx, 1 ; ecx = 1

powLoop:

cmp ecx, [ebp+8] ; m

jge endPowLoop ; Завершение цикла

fmul st1, st ; (1+x)^(ecx-1) \* (1+x)

inc ecx ; ecx++

jmp powLoop ; Возвращаемся в начало цикла

endPowLoop:

fmulp st1, st

add esp, 4

pop ebp

ret

; ---------------------------------------------------------------------------------------

section '.idata' import data readable

library kernel,'kernel32.dll',\

user,'user32.dll',\

msvcrt,'msvcrt.dll'

import kernel,\

ExitProcess,'ExitProcess'

import msvcrt,\

sscanf,'sscanf',\

gets,'gets',\

\_getch,'\_getch',\

printf,'printf',\

scanf,'scanf'

**Тестирование**

При вводе некорректных данных при чтении значения программа выводит сообщение “Wrong number.” и просит пользователя ввести значение еще раз. В случае если пользователь передал некорректные данные при чтении параметра программа выводит сообщение “Wrong number.” после чего завершает работу программы (см. рис. 1).

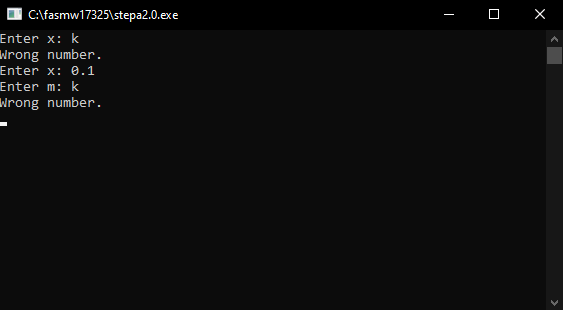
**

Рисунок 1 – Обработка некорректных входных данных

Если все введены корректные данные, то программа подсчитывает значение ряда Тейлора с точностью 0.05% (0.0005) и точное значение функции и выводит их на экран (см. рис. 2)

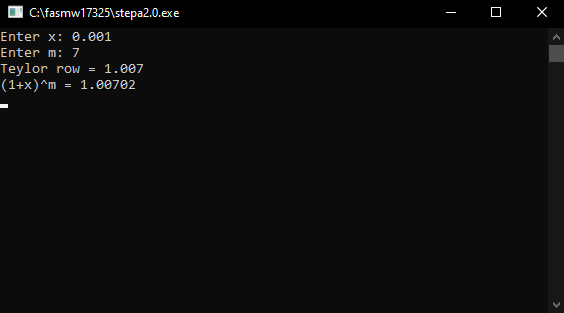


Рисунок 2 – Результат работы программы при вводе корректных данных

**Список используемых источников**

1. Википедия (2020) «Ряд Тейлора» ([https://ru.wikipedia.org/wiki/Ряд\_Тейлора)](https://ru.wikipedia.org/wiki/Ряд_Тейлора)
2. Легалов А.И.(2020) «Разработка программ на ассемблере. Использование макроопределений» ([http://softcraft.ru/edu/comparch/practice/asm86/04-macro/)](http://softcraft.ru/edu/comparch/practice/asm86/04-macro/)
3. Легалов А.И.(2020) «Разработка программ на ассемблере. Использование сопроцессора с плавающей точкой» [(http://softcraft.ru/edu/comparch/practice/asm86/05-fpu/)](http://softcraft.ru/edu/comparch/practice/asm86/05-fpu/)
4. SoftCraft разноликое программирование «Программирование на языке ассемблера. Микропроект. Требования к оформлению. 2020-2021 уч.г.» (<http://softcraft.ru/edu/comparch/tasks/mp01/>)
5. natalia.appmat «Программирование на языке ассемблера» (<http://natalia.appmat.ru/c&c++/assembler.html>)
6. osinavi «Команды передачи управления» (<http://osinavi.ru/asm/4.html>)
7. osinavi «FPU (Floating Point Unit)» (<http://osinavi.ru/asm/FPUexpansion/1.html>)
8. osinavi « Основные команды загрузки и сохранения» (<http://osinavi.ru/asm/FPUexpansion/2.html>)
9. osinavi « Арифметические команды сопроцессора» (<http://osinavi.ru/asm/FPUexpansion/3.html>)
10. osinavi «Математические команды сопроцессора» (<http://osinavi.ru/asm/FPUexpansion/4.html>)
11. osinavi « Команды сравнения FPU» (<http://osinavi.ru/asm/FPUexpansion/5.html>)
12. osinavi «Команды управления FPU» (<http://osinavi.ru/asm/FPUexpansion/6.html>)
13. osinavi « Дополнительные возможности сопроцессора» (<http://osinavi.ru/asm/FPUexpansion/7.html>)