СОГЛАСОВАНО

Научный руководитель,  
доцент департамента программной инженерии факультета компьютерных, канд. техн. наук

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Легалов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

УТВЕРЖДАЮ

**Микропроект**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 81 01-1 ЛУ**

Исполнитель

cтудент группы БПИ197  
\_\_\_\_\_\_\_\_\_\_\_\_\_/ П. О. Кулешова /

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В. В. Шилов

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

УТВЕРЖДЕН   
RU.17701729.04.01-01 81 01-1

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

**Микропроект**

**Пояснительная записка**

**RU.17701729.04.01-01 81 01-1**

**Листов** 24

сОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc54972482)

[1.1. Наименование программы 3](#_Toc54972483)

[1.2. Документ, на основе которого ведется разработка 3](#_Toc54972484)

[1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 4](#_Toc54972485)

[1.1 Назначение разработки 4](#_Toc54972486)

[1.2 Краткая характеристика области применения 4](#_Toc54972487)

[2. Описание программы 5](#_Toc54972488)

[2.1 Условие задачи 5](#_Toc54972489)

[2.2 Метод решения 5](#_Toc54972490)

[2.3 Алгоритм решения 5](#_Toc54972491)

[Приложение 1 6](#_Toc54972492)

[3. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 7](#_Toc54972493)

[ПРИЛОЖЕНИЕ 1 ТАБЛИЦЫ С ОПИСАНИЕМ МЕТОДОВ 8](#_Toc54972494)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 11](#_Toc54972495)

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы: Микропроект

Краткое наименование программы: Микропроект.

## Документ, на основе которого ведется разработка

http://softcraft.ru/edu/comparch/tasks/mp01/#m7

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## Назначение разработки

Назначение разработки – вычисление на языке Ассемблер, в среде разработки FASM.

## Краткая характеристика области применения

Научно-исследовательская область.

# Описание программы

## Условие задачи

Разработать программу, вычисляющую спомощью степенного ряда с точностью нехуже 0,1% значение функции длязаданного параметра x (использовать FPU)

## Область допустимых значений

[-10;10], с точностью до шести знаков после запятой

## Метод решения

Задача решалась с применением метода разложения в ряд Тейлора:

## Алгоритм решения

1. Производим ввод переменной
2. В цикле, пока разница между текущей и предыдущей значениями сумм не будет равна нулю (то есть n-й член ряда Тейлора не станет равным машинному нулю):
   1. Мы высчитываем i-й член ряда Тейлора
   2. Прибавляем его к уже имеющейся сумме
   3. Проверяем на ноль, если ноль – выходим, иначе – повторяем пункт 2
3. Высчитываем погрешность как разность текущей и предпредыдущей суммы (так как значение предыдущей суммы = значению текущей по условию)
4. Выводим значение функции и погрешности

# ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению// Единая система программной документации. – М.: ИПК Стандартинформ, 2010.
2. Сайт «SoftCraft». URL: http://www.softcraft.ru/edu/comparch/practice/asm86/05-fpu/sqroot32/sqroot32.asm [http://www.softcraft.ru] Просмотрено: 27.10.2020
3. Сайт «Программирование». URL: https://prog-cpp.ru/asm-coprocessor-command/ Просмотрено: 27.10.2020
4. Сайт «FPUexpansion». URL: http://osinavi.ru/asm/FPUexpansion/1.html Просмотрено: 30.10.2020
5. Сайт «Программирование на языке Ассемблера». URL: <http://natalia.appmat.ru/c&c++/assembler.html> Просмотрено: 30.10.2020

# ПРИЛОЖЕНИЕ 1 **Таблицы с описанием команд**

**Таблица 1.** Описание команд

|  |  |  |
| --- | --- | --- |
| имя | аргументы | назначение |
| fld | src | Загрузка значения в вершину стека |
| fmulp | - | Умножение верхних элементов стека с выталкиванием |
| fst | src | Запись значения вершины стека в переменную |
| fstp | src | Запись с выталкивание значения вершины стека в переменную |
| fdivp | - | Деление верхних элементов стека с выталкиванием |
| fcom | src | Вещественное сравнение St(0)-src, где St(0) – значение вершины стека |
| fstsw | ax | Сохранение слова-состояния в регистр ax |
| jz | label | Переход по метки если ZF=1 |
| jmp | label | Команда безусловного перехода |
| faddp | - | Сложение верхних элементов стека с выталкиванием |
| fld1 | - | Загрузка в вершину стека единицы |
| fchs | - | Инвертирует знак переменной в вершине стека |
| fsubp | src | Вычитание верхних элементов стека с выталкиванием |
| fabs | - | Взятие модуля от числа в вершине стека |

# ПРИЛОЖЕНИЕ 2 **Код программы**

format PE console

entry start

include 'win32a.inc'

section '.code' code readable executable

start:

***;Ввод значений***

invoke printf, yValue ;y=1/(e^x)

invoke printf, xValue ;x=

invoke scanf, istr, x ;input x

FINIT

***;Ряд тейлора (цикл)***

;teylor series

taylor:

***;Загружаем на стек (-1)^n***

fld qword[a] ;sign a=(-1)^n

***;Загружаем на стек предыдущее значение факториала***

fld qword[lastnum] ;lastnum=(n-1)!

;***Загружаем на стек n***

fld qword[n] ;n

;***Вычислем новое значение факториала***

fmulp ;n\*(n-1)!

***;Записываем новое значение факториала в переменную***

fst qword[lastnum] ;lastnum=n!

***;Находим частное(-1)^n/n!***

fdivp ;(-1)^n/n!

***;Загружаем на стек x^(n-1)***

fld qword[lastx] ;lastx=x^(n-1)

;***Загружаем на стек х***

fld qword[x] ;x

;***Вычисляем значение x^n***

fmulp ;x\*x^(n-1)

***;Запоминаем это значение в переменной***

fst qword[lastx] ;lastx=x^n

***;Находим n-й элемент ряда Тейлора***

fmulp ;x^n\*(-1)^n/n!

***;Загружаем уже имеющуюся сумму ряда Тейлора***

fld qword[sum] ;put sum

***;Проверяем равна ли она предыдущему значению***

fcom qword[sumPrevious] ;real comparison

***;Переписываем значение SR регистра в AX***

fstsw ax ;saves the current value of the SR register

***;Копируем значения AH регистра в регистр флагов***

sahf ;copy the contents of the AH register into the lower 8 bits of the flags register

***;Выгружаем сумму***

fstp qword[sum] ;unload sum

***;Если мы достигли желаемой точности (машинного нуля), то есть ;текущие значение == предыдущему, то мы выходим***

jz .end ;if ZF=1, so sum==sumPrevious

***;Записываем предыдущую сумму***

fld qword[sumPrevious]

***;Теперь она предпредыдущая***

fstp qword[sumPreviousPrevious]

***;Загружаем сумму***

fld qword[sum] ; teylor sum from n-1

***;Теперь она предыдущая сумма***

fst qword[sumPrevious] ;save previous sum

***;Прибавляем к ней новый член ряда***

faddp ;plus step from n

***;Выгружаем новое значение суммы***

fstp qword[sum] ;sum=sum+x^n\*(-1)^n/n!

***;Загружаем текущие значение (-1)^n***

fld qword[a] ;a=(-1)^n

***;Загружаем единицу***

fld1 ;1

***;Меняем ей знак на -1***

fchs ;-1 (sign change)

***;Перемножаем, получая (-1)\*(-1)^n***

fmulp ;(-1)\*(-1)^n

***;Записываем её в нашу переменную***

fstp qword[a] ;a=(-1)^(n+1)

***;Записываем n***

fld qword[n] ;n

***;Записываем 1***

fld1 ;1

***;Увеличиваем n на единицу***

faddp ;n+1

***;Записываем новое значение n***

fstp qword[n] ; n=n+1

***;Заходим на новый цикл***

jmp taylor ;cycle

***;Окончание работы программы***

.end:

***;Записываем сумму***

fld qword[sum] ;previous sum

***;Записываем предпредыдущие значение***

fld qword[sumPreviousPrevious];difference

***;Считаем разницу***

fsubp;

***;Берём модуль от неё***

fabs ; module

***;Записываем её в переменную ошибки***

fstp qword[error] ;error

***;Выводим результат работы программы***

invoke printf, ostr, dword[sum], dword[sum+4]

invoke printf, errorStr, dword[error], dword[error+4]

invoke printf, previousStr, dword[sumPreviousPrevious], dword[sumPreviousPrevious+4]

***;Завершаем работу программы***

ret

section '.data' data readable writable

a dq -1.0 ;sign

x dq 0.0 ;value

lastx dq 1.0 ; previous x power

lastnum dq 1.0;previous factorial value

n dq 1.0 ;iterator

yValue db 'y=1/(e^x)',10,0

xValue db 'x=',0

istr db '%lf',0

ostr db 'y=%.30lf',10,0 ;answer

sum dq 1.0 ;teylor series sum

sumPrevious dq 0.0 ;teylor series sum

sumPreviousPrevious dq 0.0 ;teylor series sum

error dq 0.0 ;teylor series sum

errorStr db 'error=%.30lf',10,0 ;error

previousStr db 'previous-previus sum=%.30lf',10,0 ;previous-previus sum

exceptionr db 'exception',10,0 ;previous-previus sum

section '.idata' import data readable

library kernel, 'kernel32.dll',\

msvcrt, 'msvcrt.dll'

import kernel,\

ExitProcess, 'ExitProcess'

import msvcrt,\

printf, 'printf',\

sprintf, 'sprintf',\

scanf, 'scanf'

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |