

СОГЛАСОВАНО

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

\_\_\_\_\_ А.И. Легалов

« \_\_\_\_ » \_\_\_\_\_ 2020 г.

УТВЕРЖДАЮ

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

\_\_\_\_\_ В. В. Шилов

« \_\_\_\_ » \_\_\_\_\_ 2020 г.

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

Микропроект

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.01-01 81 01-1 ЛУ

Исполнитель

студент группы БПИ197

\_\_\_\_\_ / П. О. Кулешова /

« \_\_\_\_ » \_\_\_\_\_ 2020 г.

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

**Микропроект**  
**Пояснительная записка**  
**RU.17701729.04.01-01 81 01-1**  
**Листов 16**

RU.17701729.04.01-01 81 01-1

**СОДЕРЖАНИЕ**

<b>ВВЕДЕНИЕ</b>	<b>3</b>
1.1. Наименование программы	3
1.2. Документ, на основе которого ведется разработка	3
<b>1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ</b>	<b>4</b>
1.1 Назначение разработки	4
1.2 Краткая характеристика области применения	4
<b>2. Описание программы</b>	<b>5</b>
Условие задачи	5
Область допустимых значений	5
Метод решения	5
Алгоритм решения	5
<b>ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ</b>	<b>6</b>
<b>ПРИЛОЖЕНИЕ 1</b> Таблицы с описанием команд	<b>7</b>
<b>ПРИЛОЖЕНИЕ 2</b> Код программы	<b>9</b>
<b>ПРИЛОЖЕНИЕ 3</b> Тестирование программы	<b>14</b>
<b>ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ</b>	<b>16</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ВВЕДЕНИЕ****1.1.Наименование программы**

Наименование программы: Микропроект

Краткое наименование программы: Микропроект.

**1.2.Документ, на основе которого ведется разработка**

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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

## 1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 1.1 Назначение разработки

Назначение разработки – вычисление  $\frac{1}{e^x}$  на языке Ассемблер, в среде разработки FASM.

### 1.2 Краткая характеристика области применения

Научно-исследовательская область.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 2. ОПИСАНИЕ ПРОГРАММЫ

### Условие задачи

Разработать программу, вычисляющую с помощью степенного ряда с точностью не хуже 0,1% значение функции  $\frac{1}{e^x}$  для заданного 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:  $\frac{1}{e^x} = 1 - \frac{x}{1!} + \frac{x^2}{2!} \dots = \sum_{n=0}^{\infty} \frac{(-1)^n x^n}{n!}$

### 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):
  - а. Мы высчитываем  $i$ -й член ряда Тейлора
  - б. Прибавляем его к уже имеющейся сумме
  - с. 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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**

1. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению// Единая система программной документации. – М.: ИПК Стандартиформ, 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

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

**ПРИЛОЖЕНИЕ 1****Таблицы с описанием команд****Таблица 1. Описание команд**

имя	аргументы	назначение
fld	src	Загрузка значения в вершину стека
fmlp	-	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 регистрах
jz	label	Переход по метки если ZF=1
jmp	label	Команда безусловного перехода

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



RU.17701729.04.01-01 81 01-1

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
fcomp	src	Вещественное сравнение с выталкиванием $St(0)$ -src, где $St(0)$ – значение вершины стека

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**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
;Проверка введённого числа
;Записываем число в стек
    fld qword[x]
;Проверка верхней границы
    fcom qword[right] ;check is x more than 10
    fstsw ax ;saves the current value of the SR register
    sahf ;copy the contents of the AH register into the lower
8 bits of the flags register
;Если превосходит 10, то переходим к label exception
    ja exception
;Проверка нижней границы с выталкиванием
    fcomp qword[left] ;check is x less than -10
    fstsw ax ;saves the current value of the SR register
    sahf ;copy the contents of the AH register into the lower
8 bits of the flags register
;Если меньше -10, то переходим к label exception
    jnb exception
;Ряд тейлора (цикл)
    ;teylor series
taylor:

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

```

;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
;Вычисляем 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 x
    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

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

```

;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 предыдущая
    fstp qword[sumPreviousPrevious]
;Загружаем сумму
    fld qword[sum] ; taylor 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
    fldl ;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
    fldl ;1
;Увеличиваем n на единицу
    faddp ;n+1
;Записываем новое значение n
    fstp qword[n] ; n=n+1

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

```

;Заходим на новый цикл
    jmp taylor    ;cycle
;Окончание работы программы
.end:
;Записываем сумму
    fld qword[sum] ;previous sum
;Записываем предыдущие 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]
    jmp exit
;x не находится в области допустимых значений
exception:
    invoke printf, exceptionr ;if x out of range
exit:
;Завершаем работу программы
    ret

section '.data' data readable writable
;Нижняя граница допустимых значений
    left dq -10.0 ;lower limit of range
;Верхняя граница допустимых значений
    right dq 10.0 ;upper limit of range
    a dq -1.0    ;sign

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.04.01-01 81 01-1

```

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  ;error
errorStr db 'error=%.30lf',10,0 ;error
previousStr db 'previous-previous sum=%.30lf',10,0
;previous-previous sum
exceptionr db 'exception. x should be between -10 and
10',10,0 ; сообщение об ошибке. X за пределами допустимых значений
section '.idata' import data readable
library kernel, 'kernel32.dll',\
msvcrt, 'msvcrt.dll'

import kernel,\
ExitProcess, 'ExitProcess'
import msvcrt,\
printf, 'printf',\
sprintf, 'sprintf',\
scanf, 'scanf'

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### ПРИЛОЖЕНИЕ 3

#### Тестирование программы

```
PS D:\polina\ForClasses\2_course\ABC\Project 1> .\f1.exe
y=1/(e^x)
x=0
y=1.00000000000000000000000000000000
error=1.00000000000000000000000000000000
previous-previous sum=0.00000000000000000000000000000000
```

#### Тест 1

```
PS D:\polina\ForClasses\2_course\ABC\Project 1> .\f1.exe
y=1/(e^x)
x=-1
y=2.71828182845904550000000000000000
error=0.0000000000000002664535259100376
previous-previous sum=2.71828182845904290000000000000000
```

#### Тест 2

```
PS D:\polina\ForClasses\2_course\ABC\Project 1> .\f1.exe
y=1/(e^x)
x=1
y=0.36787944117144239000000000000000
error=0.000000000000000166533453693773
previous-previous sum=0.36787944117144220000000000000000
```

#### Тест 3

```
y=1/(e^x)
x=0.0001
y=0.99990000499983334000000000000000
error=0.0000000000000166644475996236000
previous-previous sum=0.99990000499999998000000000000000
```

#### Тест 4

```
PS D:\polina\ForClasses\2_course\ABC\Project 1> .\f1.exe
y=1/(e^x)
x=-0.912
y=2.48929615041074030000000000000000
error=0.000000000000000444089209850063
previous-previous sum=2.48929615041073980000000000000000
```

#### Тест 5

```
PS D:\polina\ForClasses\2_course\ABC\Project 1> .\f1.exe
y=1/(e^x)
x=-10
y=22026.46579480671100000000000000000000
error=0.00000000000007275957614183425900
previous-previous sum=22026.46579480670300000000000000000000
PS D:\polina\ForClasses\2_course\ABC\Project 1> .
```

#### Тест 6

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## Тест 7

## Test 8

## Тест 9

## Тест 10

## Тест 11

## Тест 12

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата



This document was created with the Win2PDF "print to PDF" printer available at  
<http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>