

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

**Микропроект №1**

**Пояснительная записка**

**Листов 5**

Андреев Аркадий

БПИ 192, подгруппа 2

Вариант 1

**Москва 2020**

## 1. Текст задания

Разработать программу, вычисляющую спомощью степенного ряда с точностью нехуже 0,1% значение функции гиперболического синуса  $sh(x) = (e^x - e^{-x})/2$  для заданного параметра  $x$  (использовать FPU).

## 2. Применяемые расчетные методы

В программе мною был использован ряд тейлора для вычисления  $sh(x)$ .

$$sh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n-1}}{2n-1} + \dots, \quad |x| < \infty$$

Я накапливал результат в result и каждую итерацию создавал новый член ряда.

## 3. Список источников

- <http://softcraft.ru/edu/comparch/> «Разработка программ на ассемблере. Использование сопроцессора с плавающей точкой»
- <https://planetcalc.ru/1116/> онлайн калькулятор для сравнения
- <http://flatassembler.narod.ru/fasm.htm> подробная документация

## 4. Результаты работы

- $x = 0$

```
x?: 0
sh(x) = (e^x - e^(-x))/2
sh(x) = x + x^3/3! + x^5/5!...
sh(x) = 0.000000
```

- $x = 0.5$

```
x?: 0.5
sh(x) = (e^x - e^(-x))/2
sh(x) = x + x^3/3! + x^5/5!...
sh(x) = 0.521094
```

- $x = 1$

```
x?: 1
sh(x) = (e^x - e^(-x))/2
sh(x) = x + x^3/3! + x^5/5!...
sh(x) = 1.175198
```

- $x = 2$

```
x?: 2
sh(x) = (e^x - e^(-x))/2
sh(x) = x + x^3/3! + x^5/5!...
sh(x) = 3.626808
```

- $x = -1$

```
x?: -1
sh(x) = (e^x - e^(-x))/2
sh(x) = x + x^3/3! + x^5/5!...
sh(x) = -1.175198
```

- $x = -0.5$

```
x?: -0.5
sh(x) = (e^x - e^(-x))/2
sh(x) = x + x^3/3! + x^5/5!...
sh(x) = -0.521094
```

- $x = \text{NaN}$

```
x?: sdsafd
sh(x) = (e^x - e^(-x))/2
sh(x) = x + x^3/3! + x^5/5!...
sh(x) = 0.000000
```

**Текст программы**

```

format PE Console
entry start

include 'win32a.inc'

section '.data' data readable writeable
x          dq    ?    ;user's value
result     dq    1.0  ;resulting row
prev_result dq    0.0  ;to compare accuracy
denominator dq    1.0  ;to compute factorial
numerator  dq    1.0  ;to compute x^n
value      dq    ?    ;numerator/denominator
const      dq    0.001 ;accuracy 0.1%

count      dd    ?    ;counter for factorial

enterX     db    'x?: ',0
mesEq      db    'sh(x) = (e^x - e^(-x))/2', 10,0
mesRow     db    'sh(x) = x + x^3/3! + x^5/5!...', 10,0
mesFpu     db    'sh(x) = %lf', 10,0
formatF    db    '%lf',0

section '.idata' import data readable

library kernel,'kernel32.dll',\
        msvcrt,'msvcrt.dll'

import kernel,\
        ExitProcess,'ExitProcess'

import msvcrt,\
        _getch,'_getch',\
        printf,'printf',\
        scanf,'scanf'

section '.code' code readable executable
start:
    invoke printf, enterX          ;enter entering message in console

    invoke scanf, formatF, x       ;input x
    FINIT                          ;exception checking

```

```
invoke printf, mesEq      ;enter equation
invoke printf, mesRow     ;enter row
```

```
mov ebx, 1                ;i = 1
```

```
fld qword [result]        ;first row element
fmul qword [x]             ; 1.0*x
fstp qword [result]        ;write a value and clear ST0
```

```
fld qword [denominator]   ;init denominator with x
fmul qword [x]             ; 1.0*x
fstp qword [denominator]   ;write a value and clear ST0
```

```
jmp looping               ;start loop
```

checking:

```
fld qword [result]        ;load result
fsub qword [prev_result]   ;result-prev_result
fdiv qword [prev_result]   ;(result-prev_result)/prev_result
```

```
fcomp qword [const]       ;compare (result-prev_result)/prev_result with
0.0001
```

```
fstsw ax;                 ;write the coprocessor flags to ax
sahf;                     ;move them to the processor flags
jb end_work               ;(result-prev_result)/prev_result <= 0.0001
```

```
jmp looping               ;else continue
```

looping:

```
fld qword [result]        ;load result
fstp qword [prev_result]   ;write it previous result to make comparison in
future
```

```
fld qword [denominator]   ;load denominator
fmul qword [x]             ;denominator*x (denominator = x in inition)
fmul qword [x]             ;denominator*x*x
fstp qword [denominator]   ;denominator=denominator*x*x and clear ST0
```

```
fld qword [numerator]     ;load numerator (1)
inc ebx                   ;i++
mov [count],ebx           ;write ebx in count
fimul dword [count]        ;numerator * count
```

```
inc ebx                   ;i++
```

```

mov [count],ebx           ;write ebx in count
fimul dword [count]       ;numerator * count
fstp qword [numerator]    ;numerator=i! and clear ST0

fld qword [denominator]   ;load denominator
fdiv qword [numerator]    ;denominator/numerator
fstp qword [value]        ;value = denominator/numerator
fld qword [result]        ;load current result
fadd qword [value]        ;resultl + value
fstp qword [result]       ;result = resultl + value and clear ST0

jmp checking              ;check to continue

```

end\_work:

```

invoke printf, mesFpu, dword[result], dword[result+4] ;output result
invoke _getch           ;waiting user's press
invoke ExitProcess, 0   ;exit

```