

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

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

профессор департамента программной инженерии
факультета компьютерных наук
Легалов Александр Иванович

**МикроПроект по предмету Архитектура Вычисленных
Системы**

Пояснительная записка

Исполнитель: студент группы _БПИ197
Ф.И: Яхя Янал

Москва 2020

Задание

Разработать программу, вычисляющую с помощью степенного ряда с точностью не хуже 0,1% значение функции гиперболического тангенса $th(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$ для заданного параметра x (использовать FPU)

применяемые расчётные методы

чтобы считать $\text{th}(x)$, через формулы $\text{th}(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$, надо использовать степенный ряд, чтобы считать e^x и e^{-x} , и это возможно через ряда Тейлора $e^x = 1 + \frac{x}{1} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$, где x вокруг нуля, но почему не использовал степенный ряд, чтобы это сработало для всех $x \in R$? Пусть $x = a$, где a большое число, и пусть $X = x - a$, тогда:

$$e^x = e^a \left(1 + \frac{(x-a)}{1} + \frac{(x-a)^2}{2!} + \dots + \frac{(x-a)^n}{n!} \right)$$

Но из начала было надо считать e^a , и вдруг опять в нашей формуле появилось ещё $f^n(e^x \text{ where } x = a) = e^a$, из этого следует что мы не можем считать e^x в нашем случае с помощью степенного ряда, поэтому в моей программе есть ограничение на x , где x должен $\in [-4.4, 4.4]$, чтобы получить ответ с точностью не хуже 0,1%.

Но а если мы взяли N какое то большое число то получим хорошую точность для больших x ? Да получим, но, а если $x=100$,

Тогда N должно быть тоже достаточно большое, пусть для этого достаточно взять $N=23$, получится что надо в ряде Тейлора считать 100^{23} , но нас $100^{23} > 2^{80} - 1$, то есть 100^{23} не поместится в директиве DT(10 bytes), поэтому увлечение N не решает нашу проблему, и вот из-за чего я ограничил значение x .

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- Презентация семинара по FPU и его запись
- Знание ряды из первого курса

Текст Программы

Microproject.ASM:

```
format PE console
entry start

include 'win32a.inc'
include 'my_lib.inc' ; including my Library

section '.code' code readable executable

start:
    ; input x
    invoke printf, strx
    invoke scanf, formatEnterX, x;

    FINIT ;coprocessor initialization

FirstCon:
    ;check if x is in range [-4.4, 4.4]
    fld [x] ;loading x into the top of the stack st(0)
    fcomp [rLimitation] ; comparing st(0) and rLimitation, with
clearing st(0)
    fstsw ax
    sahf
    jnbe start ; if st(0) > rLimitation

secondCon:
    fld [x]
    fcomp [lLimitation] ; comparing st(0) and lLimitation, with
clearing st(0)
    fstsw ax
    sahf
    jb start ; if st(0) < lLimitation

    invoke printf, formatX, dword[x], dword[x+4] ;print x
    invoke printf, formatInt ,[N]

    TH [x] ; calling TH macro in my_lib

    ; printing the result of tanh(x)
    invoke printf, formatTH,dword[x], dword[x+4], dword[th_res], dword[th_res+4]
    invoke getch

;-----third act - including HeapApi-----
-
section '.data' data readable writable

    strx db "Enter x between [-4.4, 4.4], to calculate tanh(x): ", 0
    formatEnterX db '%lf' , 0
    formatTH db 'tanh(%lf) = %lf',10, 0
    formatX db 'x = %lf',10, 0
    formatInt db 'N = %d',10, 0
    x dq 0
    e_To_X dq ?
```

```

e_To_nX      dq      ?
N            dd      12
resPow       dq      ?
resFac       dd      ?
resSum       dq      1.0
j            dd      0
th_res       dq      ?
denominator  dq      ?
rLimitation  dq      4.4
lLimitation  dq      -4.4
tmp          dq      ?
section '.idata' import data readable
library kernel, 'kernel32.dll',\
          msvcrt, 'msvcrt.dll'

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

```

my_lib.inc:

```

;-----Factorial macro-----
macro Fac N {
local Factorial, endf
    mov eax, 1                ; loading 1 into register eax, in eax will be saved
the result of N!
    mov ecx, N                ; loading N into register ecx
Factorial:
    cmp ecx,0                 ;compare ecx with 0
    jle endf                  ;end loop if N <= 0
    mul ecx                   ; eax <- eax * ecx
    dec ecx                   ; ecx <- ecx - 1
    jmp Factorial             ; goto Factorial
endf:
    push eax                  ; pushing the value from eax to the stack
}

;-----Power macro-----
macro Power X, N{
local CalcPower, endPowerer
    mov eax, 1                ; loading 1 into register eax, in eax will be saved
the result of N!
    mov ecx, N                ; loading N into register ecx
    fld1                       ; loading 1 into the top of the stack st(0)
CalcPower:
    cmp ecx, 0                 ; comparing ecx with 0
    je endPowerer              ; end loop if ecx == 0
    fmul X                     ; st(0) <- st(0) * X
    dec ecx                   ; ecx <- ecx - 1
    jmp CalcPower              ; goto CalcPower
endPowerer:
    fstp [resPow]              ; writing the result of x^N to memory in resPow, with
clearing the stack

```

```

}

;-----e^x macro-----
macro EpowX X, N{
local Ex_Loop, Start_Ex_Loop, end_Ex_Loop
Ex_Loop:
    mov ebx, N                ; loading N into register ebx
Start_Ex_Loop:
    cmp ebx, 0                ; compare ebx with 0
    je end_Ex_Loop            ; end the loop if ebx == 0
    mov [j], ebx               ; saving temp value of ebx in j
    Fac ebx                    ; calling Fac macro to calculate ebx!
    pop [resFac]               ; saving the result of Fac macro(ebx!) which was
pushed to stack, in memory into resFac
    Power X, ebx               ; calling Power macro to calculate X^ebx
    fld [resPow]               ; loading resPow into the top of the stack st(0)
    fdiv [resFac]               ; s(0) <- s(0) / resFac
    fadd [resSum]               ; s(0) <- s(0) + resSum
    fstp [resSum]              ; resSum <- s(0), and clear s(0)
    mov ebx, [j]               ; restore the value of ebx from the value of j
    dec ebx                    ; ebx <- ebx - 1
    jmp Start_Ex_Loop          ; goto Start_Ex_Loop
end_Ex_Loop:
    nop                        ; no operation
}

;-----tanh(x) macro-----
macro TH X{
    fld X                       ; loading X into the top of the stack st(0)
    fstp [tmp]                  ; tmp <- st(0), and clear st(0)
    EpowX X, [N]                ; calling EpowX macro (e^X)
    fld [resSum]                ; loading resSum into the top of the stack st(0)
    fstp [e_To_X]               ; e_To_X <- st(0), and clearing st(0)
    ; reinitialize resSum to 1
    fld1                        ; loading 1 into the top of the stack st(0)
    fstp [resSum]               ; resSum <- st(0), and clearing st(0)
    ; getting -X
    fldz                        ; loading 0 into the top of the stack st(0)
    fsub X                       ; st(0) <- st(0) - X
    fstp X                       ; X <- st(0), and clear st(0)

    EpowX X, [N]                ; calling EpowX macro (e^-X)
    fld [resSum]                ; loading resSum into the top of the stack st(0)
    fstp [e_To_nX]              ; e_To_X <- st(0), and clearing st(0)
    fld [e_To_X]                ; loading e_To_X into the top of the stack st(0)
    fadd [e_To_nX]               ; st(0) <- st(0) + e_To_nX
    fstp [denominator]          ; denominator <- st(0), and clear st(0)
    fld [e_To_X]                ; loading e_To_X into the top of the stack st(0)
    fsub [e_To_nX]               ; st(0) <- st(0) - e_To_nX
    fdiv [denominator]           ; st(0) <- st(0) / denominator
    fstp [th_res]               ; th_res <- st(0), and clear st(0)
    ; re-assign to base its base value
    fld [tmp]
    fstp X
}

```

macro Fac:

	Входные параметры	Выходные параметры
Имя	N	закружится в стек
Тип	dd (4 bytes)	dd (4 bytes)
Назначение	Число, которое будет вычислено его факториал	N!
Метод передачи	По значению	По значению в стек
Функциональность макроса	Посчитать факториал определённого числа	

Macro Power:

	Входные параметры	Выходные параметры
Имя	X, N	resPow
Тип	dq (8 bytes), dd(4bytes)	dq (8bytes)
Назначение	X: число для которого будет вычислено tanh N: степень X	X^N
Метод передачи	По значению	По значению из регистра FPU в памяти (resPow)
Функциональность макроса	Посчитать X^N где, X вещественное число и N целое положительное число	

Macro EpowX:

	Входные параметры	Выходные параметры
Имя	X, N	resSum
Тип	dq (8 bytes), dd(4bytes)	dq (8bytes)
Назначение	X: число для которого будет вычислено tanh N: то N которое в формуле ряда Тейлора	e^X
Метод передачи	По значению	По значению из регистра FPU в памяти (resSum)
Функциональность макроста	Посчитать e^X где, X вещественное число	

Macro TH:

	Входные параметры:	Выходные параметры:
Имя	X	th_res
Тип	dq (8 bytes)	dq (8bytes)
Назначение	X: число для которого будет вычислено tanh	Tanh(x)
Метод передачи	По значению	По значению из регистра FPU в памяти (th_res)
Функциональность макроста	Посчитать $\tanh(X)$ где, X вещественное число	

Примеры Выполнение

```
E:\Second Year HSE\AVS\mikroproject\microProject.EXE
Enter x between [-4.4, 4.4], to calculate tanh(x): 4.5
Enter x between [-4.4, 4.4], to calculate tanh(x): 4.6
Enter x between [-4.4, 4.4], to calculate tanh(x): -4.5
Enter x between [-4.4, 4.4], to calculate tanh(x): -4.6
Enter x between [-4.4, 4.4], to calculate tanh(x): 4.4
x = 4.400000
N = 12
tanh(4.400000) = 0.999006
```

```
E:\Second Year HSE\AVS\mikroproject\microProject.EXE
Enter x between [-4.4, 4.4], to calculate tanh(x): 0.5
x = 0.500000
N = 12
tanh(0.500000) = 0.462117
```

E:\Second Year HSE\AVS\mikroproject\microProject.EXE

```
Enter x between [-4.4, 4.4], to calculate tanh(x): 1.7
x = 1.700000
N = 12
tanh(1.700000) = 0.935409
-
```

E:\Second Year HSE\AVS\mikroproject\microProject.EXE

```
Enter x between [-4.4, 4.4], to calculate tanh(x): -3.7865
x = -3.786500
N = 12
tanh(-3.786500) = -0.998784
-
```

E:\Second Year HSE\AVS\mikroproject\microProject.EXE

```
Enter x between [-4.4, 4.4], to calculate tanh(x): -0.348726
x = -0.348726
N = 12
tanh(-0.348726) = -0.335245
```

E:\Second Year HSE\AVS\mikroproject\microProject.EXE

```
Enter x between [-4.4, 4.4], to calculate tanh(x): 0
x = 0.000000
N = 12
tanh(0.000000) = 0.000000
```