

НИУ ВШЭ

Факультет компьютерных наук

Программная инженерия

Программа для вычисления гиперболического косинуса с помощью
степенного ряда

Куприхин Дмитрий Алексеевич, БПИ199

1. Оглавление

1. Текст программы.....	3
2. Применяемый метод	3
3. Тестовые примеры	3
Список литературы	6
Приложение 1. Текст файла hw3Macro.inc	7
Приложение 2. Текст файла hw3.asm.....	8

1. Текст программы

Вариант 14. Разработать программу, вычисляющую с помощью степенного ряда с точностью не хуже 0,05% значение функции гиперболического косинуса $\text{ch}(x) = (e^x + e^{-x})/2$ для заданного параметра x (использовать FPU).

2. Применяемый метод

Значение гиперболического косинуса для аргумента x может быть выражено через ряд Тейлора[1]:

$$\cosh(x) = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}$$

3. Тестовые примеры

Значения сравнивались с результатами программы Калькулятор

- $X = 0$. Результат Калькулятора = 1.

```
C:\Users\Дмитрий Куприхин\Desktop\FASM\hw3.EXE
This program calculates cosh(x). Enter x (-700 <= x <= 700)
0
Iteration 1: current term = 1.000000, sum = 1.000000
Iteration 2: current term = 0.000000, sum = 1.000000
Result: cosh(0.000000) = 1.000000
```

- $X = 0.01$. Результат Калькулятора = 1,000050.

```
C:\Users\Дмитрий Куприхин\Desktop\FASM\hw3.EXE
This program calculates cosh(x). Enter x (-700 <= x <= 700)
0.01
Iteration 1: current term = 1.000000, sum = 1.000000
Iteration 2: current term = 0.000050, sum = 1.000050
Result: cosh(0.010000) = 1.000050
```

- $X = 0.53$. Результат Калькулятора = 1,143768.

```
C:\Users\Дмитрий Куприхин\Desktop\FASM\hw3.EXE
This program calculates cosh(x). Enter x (-700 <= x <= 700)
0.53
Iteration 1: current term = 1.000000, sum = 1.000000
Iteration 2: current term = 0.140450, sum = 1.140450
Iteration 3: current term = 0.003288, sum = 1.143738
Iteration 4: current term = 0.000031, sum = 1.143768
Result: cosh(0.530000) = 1.143768
```

- $X = 1$. Результат Калькулятора = 1,543080.

```
C:\Users\Дмитрий Куприхин\Desktop\FASM\hw3.EXE
This program calculates cosh(x). Enter x (-700 <= x <= 700)
1
Iteration 1: current term = 1.000000, sum = 1.000000
Iteration 2: current term = 0.500000, sum = 1.500000
Iteration 3: current term = 0.041667, sum = 1.541667
Iteration 4: current term = 0.001389, sum = 1.543056
Iteration 5: current term = 0.000025, sum = 1.543080
Result: cosh(1.000000) = 1.543080
```

- $X = 12.33$. Результат Калькулятора = 113193,363811.

```
C:\Users\Дмитрий Куприхин\Desktop\FASM\hw3.EXE
This program calculates cosh(x). Enter x (-700 <= x <= 700)
12.33
Iteration 1: current term = 1.000000, sum = 1.000000
Iteration 2: current term = 76.014450, sum = 77.014450
Iteration 3: current term = 963.032768, sum = 1040.047218
Iteration 4: current term = 4880.293747, sum = 5920.340965
Iteration 5: current term = 13249.030179, sum = 19169.371143
Iteration 6: current term = 22380.394268, sum = 41549.765411
Iteration 7: current term = 25776.263046, sum = 67326.028458
Iteration 8: current term = 21531.521522, sum = 88857.549980
Iteration 9: current term = 13639.223051, sum = 102496.773031
Iteration 10: current term = 6776.327050, sum = 109273.100082
Iteration 11: current term = 2711.046178, sum = 111984.146259
Iteration 12: current term = 892.115516, sum = 112876.261775
Iteration 13: current term = 245.701704, sum = 113121.963479
Iteration 14: current term = 57.467323, sum = 113179.430801
Iteration 15: current term = 11.556473, sum = 113190.987275
Result: cosh(12.330000) = 113190.987275
```

- $X = -7.8965$. Результат Калькулятора = 1343,929356.

```
C:\Users\Дмитрий Куприхин\Desktop\FASM\hw3.EXE
This program calculates cosh(x). Enter x (-700 <= x <= 700)
-7.8965
Iteration 1: current term = 1.000000, sum = 1.000000
Iteration 2: current term = 31.177356, sum = 32.177356
Iteration 3: current term = 162.004589, sum = 194.181945
Iteration 4: current term = 336.724985, sum = 530.906930
Iteration 5: current term = 374.935527, sum = 905.842457
Iteration 6: current term = 259.766632, sum = 1165.609090
Iteration 7: current term = 122.709649, sum = 1288.318738
Iteration 8: current term = 42.041345, sum = 1330.360083
Iteration 9: current term = 10.922817, sum = 1341.282900
Iteration 10: current term = 2.225781, sum = 1343.508681
Iteration 11: current term = 0.365231, sum = 1343.873913
Result: cosh(-7.896500) = 1343.873913
```

- [illegible]

Список литературы

1. Abramowitz, M., & Stegun, I. A. (1972). *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications.

Приложение 1. Текст файла hw3Macro.inc

; Макрос для вывода чисел с плавающей запятой.

```
macro PrintFloat string, [args] {  
    reverse  
    push dword[args + 4]  
    push dword[args]  
    common  
    push string  
    call [printf]  
}
```

; Макрос, инкрементирующий значение переменной.

```
macro Inc value {  
    mov ecx, [value]  
    inc ecx  
    mov [value], ecx  
}
```

; Макрос, копирующий значение одного числа с плавающей запятой в другое.

```
macro CopyFloat to, from {  
    mov eax, dword[from]  
    mov ebx, dword[from + 4]  
    mov dword[to], eax  
    mov dword[to + 4], ebx  
}
```

Приложение 2. Текст файла hw3.asm

; Программа для вычисления гиперболического косинуса
(вариант 14) с помощью

; степенного ряда с точностью не хуже 0.05%.

; $\cosh(x) = 1 + (x^2 / 2!) + (x^4 / 4!) + \dots$

; Автор: Куприхин Дима, БПИ199

format PE Console

entry start

include 'win32a.inc'

include 'hw3Macro.inc'

section '.data' data readable writable

enterStr db 'This program calculates cosh(x). Enter
x', 10, 0

numberStr db '%lf', 0

iterationStr db 'Iteration %d: ', 0

currentValuesStr db 'current term = %lf, sum = %lf',
10, 0

resultStr db 'Result: cosh(%lf) = %lf', 10, 0

; Вводимый пользователем аргумент вычисляемой
функции

x dq ?

; Квадрат введенного значения.

sqrX dq ?

; Переменная, хранящая значение текущего члена ряда.

term dq 1.0

; Переменная, в которой вычисляется значение
следующего члена ряда.

nextTerm dq ?


```

; Счетчик значения факториала.
factorialCounter dd 1
; Счетчик итерации.
iterationCounter dd 1
; Переменная, хранящая сумму ряда.
sum dq 1.0
; Переменная, хранящая точность, с которыми
производятся вычисления.
epsilon dq 0.0005

section '.code' code readable executable

start:
    ; Вводим аргумент функции.
    invoke printf, enterStr
    invoke scanf, numberStr, x

    FINIT
    ; Считаем  $x^2$ , так как каждый следующий член
ряда в
    ; числителя увеличивается на  $x^2$ 
    fld [x]
    fmul [x]
    fstp [sqrX]

mainLoop:
    ; Выводим промежуточный результат.
    invoke printf,
iterationStr,[iterationCounter]
    PrintFloat currentValuesStr, term, sum

```

Inc iterationCounter

; Вычисляем следующий член:

; nextTerm = term; nextTerm *= x^2;

fld [term]

fmul [sqrX]

; nextTerm /= factorialCounter++;

fidiv [factorialCounter]

Inc factorialCounter

; nextTerm /= factorialCounter++;

fidiv [factorialCounter]

Inc factorialCounter

+= nextTerm;

; Прибавляем к сумме вычисленный член. sum

fstp [nextTerm]

fld [sum]

fadd [nextTerm]

fstp [sum]

; Проверяем на остановку алгоритма.

заканчиваем

; Если новый член меньше 0.05% от суммы, то

; вычисления и выводим результат.

fld [sum]

fmul [epsilon]

fld [nextTerm]

fcomp

fstsw ax

sahf

```
jb outputResult
```

```
; term = nextTerm;
```

```
CopyFloat term, nextTerm
```

```
jmp mainLoop
```

```
outputResult:
```

```
PrintFloat resultStr, x, sum
```

```
finish:
```

```
push 0
```

```
invoke scanf, numberStr, x
```

```
call ExitProcess
```

```
section '.idata' import data readable
```

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

```
msvcrt, 'msvcrt.dll'
```

```
import kernel,\
```

```
ExitProcess, 'ExitProcess',\
```

```
GetProcessHeap, 'GetProcessHeap',\
```

```
HeapAlloc, 'HeapAlloc'
```

```
import msvcrt,\
```

```
printf, 'printf',\
```

```
scanf, 'scanf'
```