ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук Департамент программной инженерии

ПРОГРАММА, ВЫЧИСЛЯЮЩАЯ С ПОМОЩЬЮ СТЕПЕННОГО РЯДА С ТОЧНОСТЬЮ НЕ ХУЖЕ 0,1% ЗНАЧЕНИЕ ФУНКЦИИ e^x ДЛЯ ЗАДАННОГО ПАРАМЕТРА x

Пояснительная записка

		Исполнитель
		студент группы БПИ193
		/С. М. Курбанова /
‹ ‹	>>	2020 г.

Оглавление

Текст задания	3
Применяемые расчётные методы	4
Обоснование области допустимых значений входных параметров	5
Гестовые примеры	6
Список использованной литературы	7
ПРИЛОЖЕНИЕ	8

Текст задания

Разработать программу, вычисляющую с помощью степенного ряда с точностью не хуже 0,1% значение функции e^x для заданного параметра x (использовать FPU).

Применяемые расчётные методы

Для нахождения значения функции $f(x) = e^x$ с точностью $\varepsilon = 0.001$ была использована формула:

$$f(x) = e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x}{2!} + \frac{x}{3!} + \dots$$

Формула нахождения каждого последующего члена:

$$x_n = x_{n-1} \times \frac{x}{n}$$

Для нахождения результата искомой функции в алгоритме к переменной result прибавляются новые члены до тех пор, пока значение последующего члена x_n превосходит или равно по модулю ε , то есть пока результат вычислений не начнёт увеличиваться на значение меньшее ε .

Рисунок 1. Фрагмент кода, соответствующий вышеописанному алгоритму.

Промежуточные вычисления располагаются в переменной result, в ней же хранится конечный результат вычислений.

Исходные данные располагаются в переменной х.

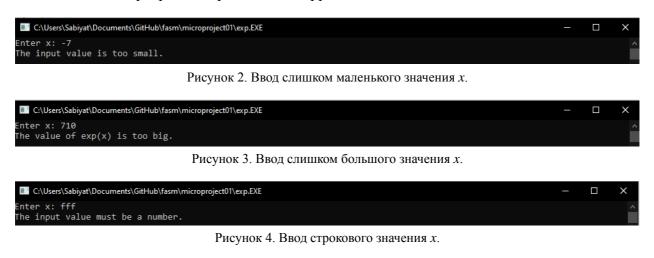
Обоснование области допустимых значений входных параметров

Минимальное допустимое значение входного параметра x равняется -6, т. к. использованная в расчетах точность $\varepsilon = 0.001$ превосходит значение функции e^x при x < -6.

Ограничение на максимальное допустимое значение входного параметра не предусмотрено, однако при переполнении в консоль выводится соответствующее сообщение (см. рисунок 3).

Тестовые примеры

1. Работа программы при вводе некорректных данных.



2. Работа программы при вводе корректных данных.

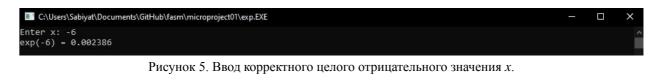




Рисунок 6. Ввод корректного вещественного отрицательного значения х.



Рисунок 7. Ввод корректного целого положительного значения x.



Рисунок 8. Ввод корректного вещественного положительного значения x.

Список использованной литературы

- 1. Инструкции FPU // [Электронный ресурс] Режим доступа: http://flatassembler.narod.ru/fasm.htm#2-1-13, свободный;
- 2. Требования к оформлению // [Электронный ресурс] Режим доступа: http://softcraft.ru/edu/comparch/tasks/mp01/, свободный;
- 3. Примеры программ // [Электронный ресурс] Режим доступа: http://softcraft.ru/edu/comparch/practice/asm86/05-fpu/, свободный.

ПРИЛОЖЕНИЕ

Текст программы

```
format PE console
    entry start
    include 'win32a.inc'
    section '.data' data readable writable
            strInput
                            db 'Enter x: ', 0
            strOutOfBounds db 'The input value is too small.',
10, 0
            strWrongFormat db 'The input value must be a
number.', 10, 0
                            db 'The value of exp(x) is too big.',
            strInfRes
10, 0
                            db '%lf', 0
            fmt
            fmtp
                            db '%lf', 13, 10, 0
                            db 'exp(%g) = %lf', 10, 0
            outputStr
                           dq?
            x
            maxL
                            dd -6.0
                                        ; result of calculation
            result
                           dq 1.0
exp(x)
                           dq 1.0
            x n
                            dd?
            tmpStack
                            dd 0
            i
                            dq 0.001
            eps
            NULL = 0
    section '.code' readable executable
            start:
            ; Gets x and checks the value
                    call Input
            ; Calcs exp(x)
                    call CalcExp
            ; Print result
                    invoke printf, outputStr, dword [x], dword
[x + 4], \setminus
                                                dword [result],
dword [result + 4]
            finish:
                    call [getch]
                    push NULL
                    call [ExitProcess]
```

```
inf:
                   push strInfRes
                   call [printf]
                   jmp finish
           outOfBounds:
                   push strOutOfBounds
                   call [printf]
                   jmp finish
           wrongFormat:
                   push strWrongFormat
                   call [printf]
                   jmp finish
           Input:
                   mov [tmpStack], esp
                   invoke printf, strInput
                   invoke scanf, fmt, x
                   cmp eax, 1
                   jne wrongFormat ; jump to wrongFormat if
input value isn't a number
                   finit
                   fld [x] ; st0 = x
fcomp [maxL] ; compare x with left border
-6.0
                   fstsw ax
                   sahf
                   jb outOfBounds ; jump to outOfBounds if x
less than -6.0
                   mov esp, [tmpStack]
                   ret
    ;-----
           CalcExp:
                   mov [tmpStack], esp
           expLoop:
           ; check if result is infinity
                   fstsw ax
                   and ax, 1000b
                   cmp ax, 0
                   jg inf
                   inc [i] ; i++
```

```
; x_n *= x / i
                   fld [x] ; st0 = x
fidiv [i] ; st0 = x / i
fmul [x_n] ; x_n *= st0
fstp [x_n] ; x_n = st0
                   ; result += x n
                   fld [x_n]; st0 = x_n
                   fabs ; st0 = |x_n|
fcomp [eps] ; compare |x_n| with 0.001
                   fstsw ax
                   sahf
                   ja expLoop ; continue loop if result
isn't accurate enough
                   mov esp, [tmpStack]
                   ret
    ;-----
   section '.idata' import data readable
           library kernel, 'kernel32.dll',\
                   msvcrt, 'msvcrt.dll'
           import kernel,\
           ExitProcess, 'ExitProcess'
           import msvcrt,\
           printf, 'printf',\
           scanf, 'scanf', \
           getch, ' getch'
```