

Пояснительная записка

ВУЗ: НИУ ВШЭ

Кафедра: Программная инженерия

Название разработки: Программа для расчёта корня 5-ой степени по быстросходящемуся итерационному алгоритму

Работу выполнил: Панфилов Егор Павлович

Группа: БПИ194

Описание

Задача: Разработать программу вычисления корня пятой степени согласно быстросходящемуся итерационному алгоритму определения корня n -ной степени с точностью не хуже 0,1% (использовать FPU).

Описание алгоритма: Алгоритм представляет из себя цикл на 200 повторений, меняющий значение переменной по упрощённой формуле быстросходящегося итерационного алгоритма. Упростить мы его можем за счёт того, что нам дан $n = 5$ (степень корня):

$$x_{k+1} = \frac{1}{n} \left((n-1)x_k + \frac{A}{x_k^{n-1}} \right) \rightarrow x_{k+1} = \frac{1}{5} \left(4x_k + \frac{A}{x_k^4} \right)$$

Все действия, за исключением ввода, вывода и хранения проводим с помощью команд FPU. За x_0 берётся A . Кол-во повторений может меняться, но взято 200 как достаточное для по нашему условию. По итогу, получившийся x_{200} и будет являться ответом.

Ссылки на источники:

<https://prog-cpp.ru/asm/>

<http://osinavi.ru/index.php?param2=18>

https://ru.wikipedia.org/wiki/Алгоритм_нахождения_корня_n-ной_степени

<http://flatassembler.narod.ru/fasm.htm#2-1-13>

Код

Далее, ниже по документу, будет предоставлен код программы. Из него, для удобства краткого ознакомления в word, убраны комментарии. В основной программе можно ознакомиться с полной версией.

```
; Program written by: Egor Panfilov, 194
; Variant: 17
; Task: Find one of the roots from the fifth root. Use fast converging iterative algorithm. Use FPU.
```

```
format PE console
```

```
entry START
```

```
include 'win32a.inc'
```

```
section '.data' data readable writable
```

```
answerOutput db 'Answer: ', 0
strInputA db 'Input A: ', 0
newLine db '', 10, 13, 0
procLF db '%lf', 0
```

```
A dq ?
xn dq ?
support dq ?
```

```
i dd 1
four dq 4.0
five dq 5.0
floatNull dq 0.0
null = 0
numberOfRepeat = 200
```

```
section '.code' code readable executable
```

```
START:
    cinvoke printf, strInputA
    cinvoke scanf, procLF, A

    fld qword [A]

    fcom [floatNull]
    fstsw ax
    sahf
    je finish

    fstp qword [xn]
```

loop1:

```
mov eax, [i]
cmp eax, numberOfRepeat
je finish
```

```
fld qword [A]
fdiv [xn]
fdiv [xn]
fdiv [xn]
fdiv [xn]
fstp qword [support]
```

```
fld qword [xn]
fmul [four]
fadd [support]
fdiv [five]
fstp qword [xn]
```

```
mov eax, [i]
add eax, 1
mov [i], eax
jmp loop1
```

finish:

```
cinvoke printf, answerOutput
invoke printf, procLF, dword[xn], dword[xn + 4]
cinvoke printf, newLine

call [getch]
push null
call [ExitProcess]
```

section '.idata' import data readable

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

```
import kernel,\
    ExitProcess, 'ExitProcess'
```

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