

Правительство Российской Федерации

**Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный
исследовательский университет «Высшая школа экономики»**

Факультет компьютерных наук Департамент программной инженерии

**Отчет к домашнему заданию по дисциплине
«Архитектура вычислительных систем»**

Работу выполнил:

Студент группы БПИ195 Скарлупин М. О.

Москва 2020

Задание

Разработать программу, вычисляющую с помощью степенного ряда с точностью не хуже 0.05% значение функции $\cos(x)$ для заданного параметра x (использовать FPU).

Решение

Будем считать косинус через разложение Маклорена:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} \dots + (-1)^n \frac{x^{2n}}{(2n)!} + o(x^{2n+1})$$

Считываем x из консоли. Инициализируем сопроцессор (FPU). Далее занесем значение x в регистр ST(0). Встроенная функция FASM'a "FCOS" вычисляет значение $\cos(x)$ и заносит результат в регистр ST(0). Затем значение ST(0) заносим в x и выводим его.

Текст программы на языке flat assembler приведен ниже:

```
;Вариант 26. Разработать программу, вычисляющую с помощью
степенного ряда с точностью не хуже 0,05% значение функции cos(x)
для заданного параметра x (использовать FPU)
```

```
;Выполнил Скарлупин Максим Олегович
```

```
format PE console ; Консольное приложение для Windows
```

```
entry start ; устанавливаем точку входа - start
```

```
include 'win32ax.inc'
```

```
;-----Variable initialization-----
```

```
section '.data' data readable writeable
```

```
strX db 'x = ', 0 ; строка для вывода в консоль
```

```
formatF      db      '%lf', 0
```

```
x dq 0.0 ; переменная для записи значения x
```

```
output db 256 dup(?) ; строка для вывода по формату
```

```
;-----Variable initialization-----
```

```
;-----Main program-----
```

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

```
start:
```

```
MainProcedure: ; основная процедура - считает значение cos(x)
```

```
    call ReadX ; считывание x
```

```
    call cosX ; вычисление cos(x). Значение находится в x
```

```
    jmp Show ; переходим на вывод результата
```

```
cosX:
```

```
    FINIT ; инициализация сопроцессора
```

```
    fld [x] ; заносим значение x в регистр ST(0)
```

```
    FCOS ; вычисляем cos(ST(0))
```

```
    FST [x] ; заносим значение косинуса в x
```

```
    ret
```

```
ReadX: ; функция считывания x
```

```
    push strX ; заносим строку "x = " в стек
```

```
call [printf] ; выводит на экран"x = "
```

```
add esp, 4
```

```
push x
```

```
push formatF
```

```
call [scanf] ; считываем x
```

```
add esp, 8
```

```
ret
```

Show: ; функция для вывода результата

```
invoke sprintf, output, 'cos(x) = %.4f', dword[x], dword[x+4]
```

; формируем строку вывода

```
call [printf] ; вызов вывода cos(x)
```

call [getch] ; вызов getch для того, чтобы консоль не
закрылась сразу

```
push 0
```

```
call [ExitProcess] ; завершение программы
```

```
;-----Main program-----  
-----
```

```
;-----Libraries-----  
-----
```

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

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

```

msvcrt, 'msvcrt.dll', \

user32, 'USER32.dll'

import kernel32,\

    ExitProcess, 'ExitProcess'

import msvcrt,\

    printf, 'printf',\

    scanf, 'scanf',\

    sprintf, 'sprintf',\

    getch, '_getch'

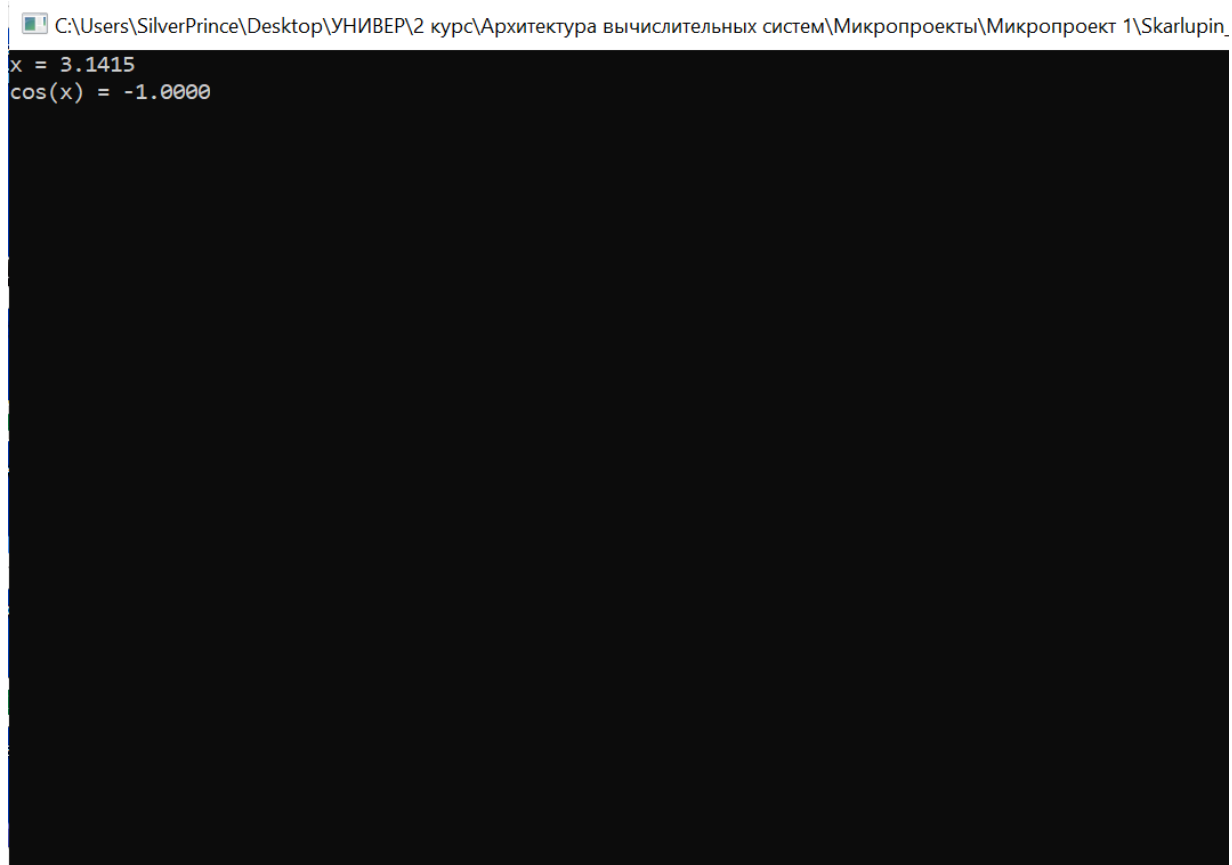
;-----Libraries-----
-----

```

Тестирование

Так как область определения для косинуса – $(-\infty, \infty)$, то некорректного ввода для числа – не может быть.

Результат выполнения программы приведен на рисунке 1.

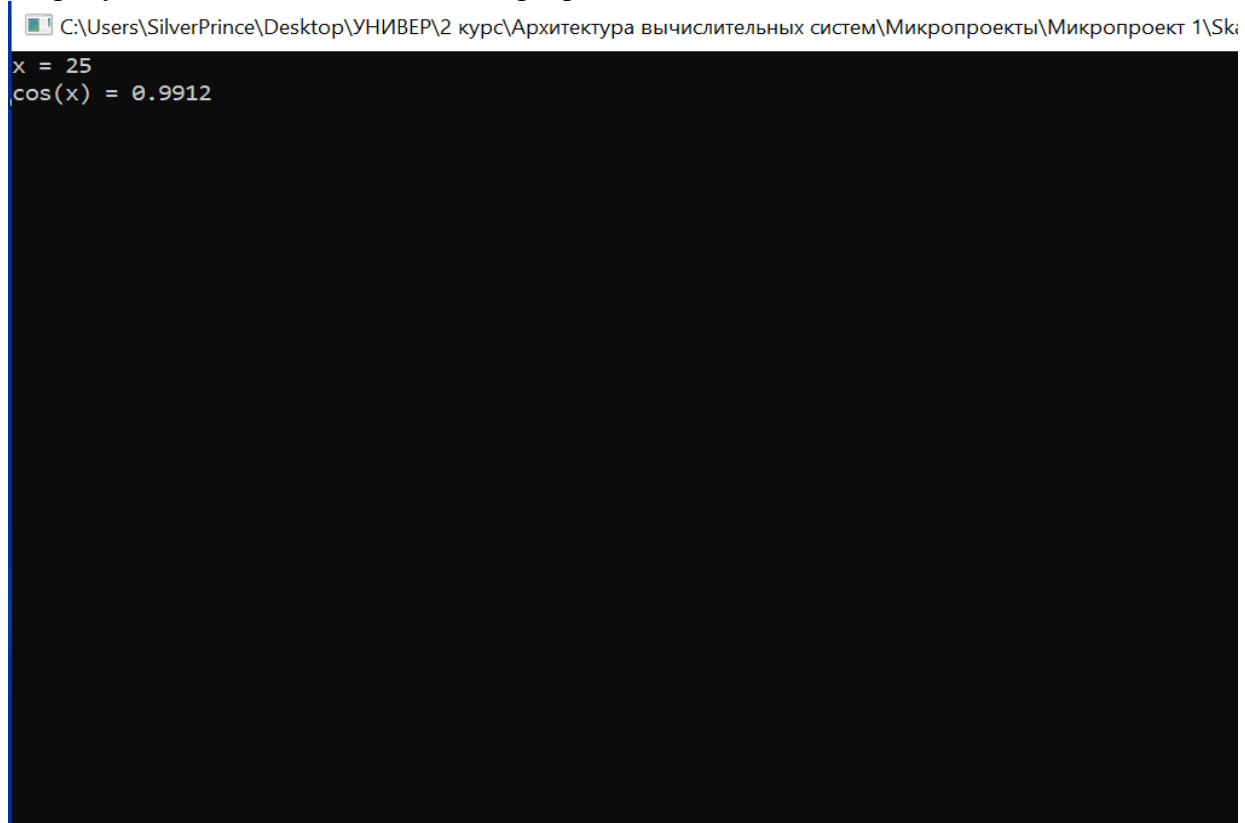


```
C:\Users\SilverPrince\Desktop\УНИВЕР\2 курс\Архитектура вычислительных систем\Микропроекты\Микропроект 1\Skarlupin_
x = 3.1415
cos(x) = -1.0000
```

Рисунок 1 – результат выполнения программы

Как видно из рисунка, погрешность меньше заданной.

На рисунке 2 показано выполнение программы для не табличного значения x .



```
C:\Users\SilverPrince\Desktop\УНИВЕР\2 курс\Архитектура вычислительных систем\Микропроекты\Микропроект 1\Skarlupin_
x = 25
cos(x) = 0.9912
```

Рисунок 2 – результат выполнения программы для не табличного значения x

Список используемых источников

1. Кип Р.И. Язык ассемблера для процессоров Intel. М.: Издательский дом “Вильямс”, 2005. – 912с

2. Легалов А.И.(2020) «Разработка программ на ассемблере. Использование макроопределений» (<http://softcraft.ru/edu/comparch/practice/asm86/04-macro/>)

Просмотрено: 30.10.2020

3. Легалов А.И.(2020) «Разработка программ на ассемблере. Использование сопроцессора с плавающей точкой» (<http://softcraft.ru/edu/comparch/practice/asm86/05-fpu/>)

Просмотрено: 31.10.2020

4. «Программирование. Система команд сопроцессора» (<https://prog-cpp.ru/asm-coprocessor-command/>)

Просмотрено 31.10.2020