**МИНОБРНАУКИ РОССИИ**

**ФЕДЕРАЛЬАНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**ИНЖЕНЕРНО-ТЕХНОЛОГИЧЕСКАЯ АКАДЕМИЯ**

**Институт компьютерных технологий и информационной безопасности Кафедра математического обеспечения и применения ЭВМ**

**Отчёт по лабораторной работе №7**

по курсу «Машинно-ориентированное программирование»

на тему: «Работа с математическим сопроцессором»

Выполнил: студент гр. КТбо2-7

Вахрушев Д.Е.

Проверил: доцент МОП ЭВМ, к.т.н.

Скороход С.В.

Таганрог 2022

# Вариант №2

**Постановка задачи**

Ввести вещественные значения a, b (a<b) и e.

Если на отрезке [a; b] функция f(x)=x3–7x2–9x+49 меняет знак на противоположный,

найти ее корень на этом отрезке методом деления отрезка пополам с точностью eps.

В противном случае вывести сообщение. Использовать двойной формат

вещественных чисел

Необходимо:

1. Разработать программу, реализующую указанную функцию.
2. Исполнить программу с несколькими наборами исходных данных, проверить правильность результатов.

# Алгоритм решения

# enter\_nums: Получение чисел a, b, eps

# Если знак функции на концах a и b не отличается

# Вывод сообщения о ошибке, переход на enter\_nums

# Запуск функции CalculateFunction(a, b, eps)

# Функция CalculateFunction (left, right, eps):

# Вычислить значения на a и b

# Если знак совпал

# Код ошибки 1, выход

# deltaX = right – left, middle = 0

# Пока (right – left > eps):

# deltaX = deltaX / 2

# middle = left + right

# Если знак f(left) не равен знаку f(right)

# Right = middle

# Иначе

# Left = middle

# Result = middle

# Сравнение выполняется «исключающим или» знаковых бит чисел

# Листинг программ

model SMALL

model SMALL

.486

stack 100h

dataseg

diff\_sign\_txt db 0Ah, 0Dh, 'Sign not different!$'

enter\_txt db 0Ah, 0Dh, 'Enter X: $'

eps\_enter\_txt db 0Ah, 0Dh, 'Enter eps: $'

result\_txt db 0Ah, 0Dh, 'Result: $'

err\_txt db 0Ah, 0Dh, 'Err! $'

str\_max\_len db 40

str\_len db ?

str\_buf db 40 dup(?)

s3coef dw 1

s2coef dw -7

s1coef dw -9

s0coef dw -10

two dw 2

number1 dq ?

number2 dq ?

number1R dq ?

number2R dq ?

eps dq ?

result dq ?

haveRes db ?

codeseg

startupcode

finit

jmp enter\_nums

not\_diff:

lea DX, [diff\_sign\_txt]

mov AH, 09h

int 21h

enter\_nums:

mov [enter\_txt + 8], 'a'

lea DX, [enter\_txt]

mov AH, 09h

int 21h

lea DX, [str\_max\_len]

mov AH, 0AH

int 21h

push offset number1

push offset str\_buf

call StrToDouble

add SP, 2

mov [enter\_txt + 8], 'b'

lea DX, [enter\_txt]

mov AH, 09h

int 21h

lea DX, [str\_max\_len]

mov AH, 0AH

int 21h

push offset number2

push offset str\_buf

call StrToDouble

add SP, 2

lea DX, [eps\_enter\_txt]

mov AH, 09h

int 21h

lea DX, [str\_max\_len]

mov AH, 0AH

int 21h

push offset eps

push offset str\_buf

call StrToDouble

add SP, 2

lea BX, [number1r]

call CalculateFunction PASCAL, number1

lea BX, [number2r]

call CalculateFunction PASCAL, number2

call DoubleSign PASCAL, number1r, number2r

jz not\_diff

call FindRoot PASCAL, number1, number2, eps

cmp [haveRes], 0

jnz ok

lea DX, [err\_txt]

mov AH, 09h

int 21h

exitcode 0

ok:

push offset str\_buf

push offset result

call DoubleToStr

add sp, 2

lea DX, [result\_txt]

mov AH, 09h

int 21h

lea DX, [str\_buf]

mov AH, 09h

int 21h

exitcode 0

;ZF = sigh difference

DoubleSign proc PASCAL @@number1:qword, @@number2:qword

uses AX

mov AL, byte ptr [@@number1+7]

xor AL, byte ptr [@@number2+7]

test AL, 080h

ret

DoubleSign endp

FindRoot proc PASCAL

arg @@left:qword, @@right:qword, @@eps:qword

local @@deltaX:qword, @@middle:qword, @@temp1:qword, @@temp2:qword

uses AX, BX

lea BX, @@temp1

call CalculateFunction PASCAL, @@left

lea BX, @@temp2

call CalculateFunction PASCAL, @@right

call DoubleSign PASCAL, @@temp1, @@temp2

@@L0:

fld @@right

fsub @@left

fstp @@deltaX

jmp @@L2

@@L4:

;deltaX /= 2;

fld @@deltaX

fild [two]

fdivp st(1), st

fstp @@deltaX

;middle = left + deltaX;

fld @@left

fadd @@deltaX

fstp @@middle

lea BX, @@temp1

call CalculateFunction PASCAL, @@left

lea BX, @@temp2

call CalculateFunction PASCAL, @@middle

call DoubleSign PASCAL, @@temp1, @@temp2

fld @@middle

jz @@L3

fstp @@right

jmp @@L2 ;to condition

@@L3:

fstp @@left

@@L2:

fld @@right

fsub @@left

fld @@eps

fcompp

fwait

fstsw AX

sahf

jb @@L4

@@L1:

fld @@middle

fstp result

mov [haveRes], 1

ret

@@err:

mov [haveRes], 0

ret

FindRoot endp

;return result on BX address

CalculateFunction proc PASCAL

ARG @@x:qword

fld @@x

fild s3coef

fmulp st(1), st

fmul @@x

fmul @@x

fld @@x

fild s2coef

fmulp st(1), st

fmul @@x

faddp st(1), st

fld @@x

fild s1coef

fmulp st(1), st

faddp st(1), st

fild s0coef

faddp st(1), st

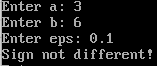
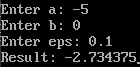
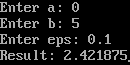
fstp qword ptr [BX]

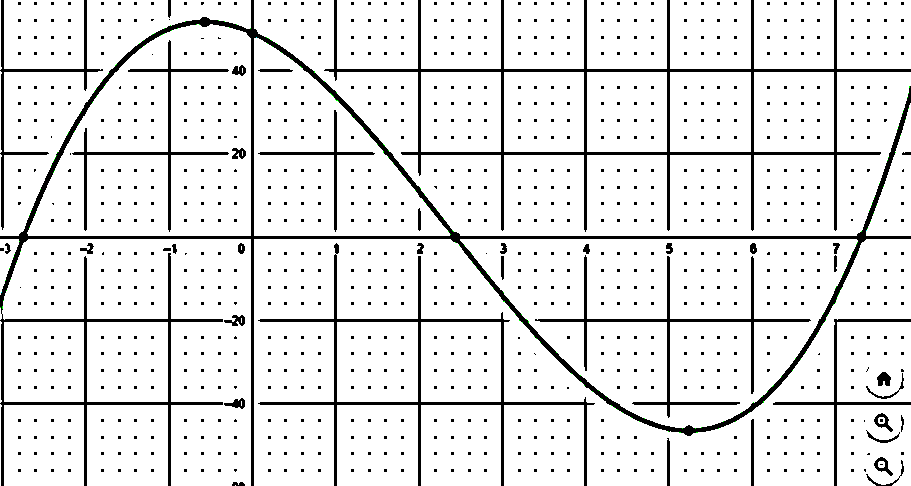
ret

CalculateFunction endp

Функции *DoubleToStr* и *StrToDouble* были взяты без изменения с Примера программы с использованием сопроцессора (стр. 76)

# Результаты тестирования

1. Пример работы программы  
   
2. Пример работы программы   
   
3. Пример работы программы  
   
4. Пример работы программы  
   



# Вывод

При выполнении лабораторной работы я получил практические навыки работы с математическим сопроцессором, а также лучше изучил работу функций в ассемблере.