**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИИ**  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**  
**(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №5

Дисциплина «Архит. ВС»

Тема «Команды сопроцессора»

Выполнил: ст. группы ПВ-31  
Ковалев Павел Александрович

Проверил: Осипов Олег Васильевич

**Белгород 2021**

**Цель работы:** изучение команд сопроцессора для выполнения арифметических операций.

**Задания для выполнения к работе**

1. Написать функцию pow (*x*, *y*) для возведения числа *x* в степень *y*. Числа *x*, *y* могут быть произвольными, в том числе отрицательными. Рассмотреть случаи, когда  
    *x* = 0 и/или *y* = 0. Аргументы передавать подпрограмме через стек. Если алгоритм требует выгрузки чисел из сопроцессора в память или регистры, использовать для этого стек. Подобрать набор тестовых данных для проверки работы функции pow (не менее 10). Убедиться в том, что результаты работы написанной функции pow и стандартной функции pow библиотеки math.h языка C или функции Math.Pow языка C# совпадают. В отчёт включить текст программы, блок-схему алгоритма функции pow и набор тестовых данных.
2. Численно исследовать на сходимость ряд. Аргументы тригонометрических функций считать в радианах. Для возведения чисел в степень использовать написанную функцию pow. В отчёт включить текст программы и значения суммы ряда при *n* от 1 до 50. Вывести результат на экран в виде:

n = 1; S = …

n = 2; S = …

…

**Выполнение работы**

|  |  |  |
| --- | --- | --- |
| Возводимая степень | Число | Результат |
| 7 | 15.785882283611476 | 244276685.97512513 |
| 4 | 12.397380786333002 | 23622.16850557444 |
| 5 | 1.5547164190911074 | 9.083556575710105 |
| 10 | 8.152607336313801 | 1297075493.1452634 |
| 2 | 8.149783655406079 | 66.41897362992407 |
| 8 | 10.432880896495366 | 140356973.6460661 |
| 8 | 2.98295215209596 | 6268.596229167466 |
| 6 | 1.538308962780702 | 13.251389593189709 |
| 3 | 1.6246359823133498 | 4.288132568321104 |

.386

.model flat, stdcall

option casemap: none

include c:\masm32\include\windows.inc

include c:\masm32\include\kernel32.inc

include c:\masm32\include\user32.inc

include c:\masm32\include\msvcrt.inc

includelib c:\masm32\lib\user32.lib

includelib c:\masm32\lib\kernel32.lib

includelib c:\masm32\lib\msvcrt.lib

.DATA

x dd 2

y dd 10

format db "n = %d, val = %G",13, 10, 0

.CODE

pow proc

;4 байта аргумент num

;4 байта аргумент pow

;4 байта - адрес возврата

;12 байт - сохранение состояний регистров

;4 байта - буфер в стеке, для передачи в сопроцессор

push ecx

push edx

push ebx

sub esp,4

mov eax,[esp+24]

mov [esp],eax

fild dword ptr [esp]

mov ecx,[esp+20]

dec ecx

pow\_cycle:

fimul dword ptr [esp]

loop pow\_cycle

fistp dword ptr [esp]

mov eax,[esp]

mov ebx,eax

add esp,4

mov eax,ebx

pop ebx

pop edx

pop ecx

ret 8

pow endp

;Принимает n

;EAX результат

row proc

push ecx

mov eax, [esp+8]

sub esp,4

mov [esp],eax

;f1

fild dword ptr [esp]

mov eax,2

mov [esp], eax

fimul dword ptr [esp]

mov eax,1

mov [esp], eax

fiadd dword ptr [esp]

;f2

mov eax, [esp+12]

mov [esp],eax

fild dword ptr [esp]

fimul dword ptr [esp]

mov eax,2

mov [esp], eax

fimul dword ptr [esp]

mov eax,1

mov [esp], eax

fisub dword ptr [esp]

;f3

fdiv st(1),st(0)

fstp dword ptr [esp]

mov eax,1

mov [esp], eax

fild dword ptr [esp]

fsubr st(1),st(0)

;f4

fstp dword ptr [esp]

mov eax, [esp+12]

mov ecx,2

mul ecx

mov ecx,eax

fldpi

fldpi

loop\_start:

dec ecx

fmul st(1),st(0)

cmp ecx,0

jne loop\_start

fmulp st(1),st(0)

;f5

mov eax,1

mov [esp], eax

fild dword ptr [esp]

fdivr st(1),st(0)

fstp dword ptr [esp]

fadd st(0),st(1)

fstp dword ptr [esp]

mov eax,[esp]

fstp dword ptr [esp]

add esp,4

pop ecx

ret 8

row endp

start:

xor eax,eax

xor ebx,ebx

xor edx,edx

xor ecx,ecx

loop\_start:

inc ecx

push ecx

call row

push eax

push 0

push ecx

push offset format

call crt\_printf

add esp,4

pop ecx

cmp ecx,50

jl loop\_start

END start