INCLUDE Irvine32.inc

ShellExecuteA PROTO, hwnd:DWORD, op:DWORD, file:DWORD, param:DWORD, dir:DWORD, show:DWORD

SW\_SHOWNORMAL equ 1

.data

space byte "-------------------------------------",0

calculator byte " COAL SCIENTIFIC CALCULATOR PROJECT ",0

made byte " made by: 21k-3107 , 21k-3063 and 21k-3073 ",0

prompt1 byte "Enter a number: ", 0

prompt2 byte "Enter an operation: 1 for addition, 2 for subtraction, 3 for multiplication, 4 for division, 5 for modulus, 6 for square, 7 for exponentiation, 8 for XOR, 9 for NOT, 10 for OR, 11 for AND, 12 for Cube, 13 for trigon, 0 to calculate): ",0

operation DWORD ?

prompt4 byte "result: ",0

prompt3 byte "press 1 to continue: press 2 to exit: ",0

thankyou1 byte "THANKS FOR USING THIS CALCULATOR ",0

thankyou2 byte "HOPE SO IT MET YOUR REQUIREMENTS SUCCESSFULLY !! GOODBYE COAL",0

cProgram db "C:\\Users\\Abdul\\c codes\\p1.exe", 0

.code

main PROC

mov eax,yellow+(black\*16)

call settextcolor

mov edx, offset space

call writestring

mov edx, offset calculator

call writestring

mov edx, offset space

call writestring

call crlf

call crlf

mov edx, offset space

call writestring

mov edx,offset made

call writestring

mov edx, offset space

call writestring

call crlf

call crlf

mov eax, green+(black\*16)

call settextcolor

callagain:

call crlf

mov eax,green+(black\*16)

mov edx, offset prompt1

call WriteString

call ReadInt

mov ecx, eax

call crlf

get\_operation:

call crlf

mov eax,green+(black\*16)

call settextcolor

mov edx, offset prompt2

call WriteString

call ReadInt

mov operation, eax

call crlf

cmp operation, 0

je done

cmp operation, 6

je square

cmp operation, 9

je nott

cmp operation, 12

je cube

cmp operation, 13

je trigonometric

jmp get\_number

square:

mov eax, ecx

imul ecx, eax

jmp get\_operation

cube:

mov eax, ecx

imul ecx, eax

imul ecx, eax

jmp get\_operation

nott:

neg ecx

jmp get\_operation

trigonometric:

invoke ShellExecuteA, NULL, NULL, ADDR cProgram, NULL, NULL, SW\_SHOWNORMAL

jmp get\_operation

get\_number:

mov edx, offset prompt1

call WriteString

call ReadInt

mov ebx, eax

cmp operation, 1

je addd

cmp operation, 2

je subtract

cmp operation, 3

je multiply

cmp operation, 4

je divide

cmp operation, 5

je modulus

cmp operation, 7

je exponentiation

cmp operation, 8

je xorr

cmp operation, 10

je orr

cmp operation, 11

je andd

addd:

add ecx, ebx

jmp get\_operation

subtract:

sub ecx, ebx

jmp get\_operation

multiply:

mov eax, ecx

imul eax, ebx

mov ecx, eax

jmp get\_operation

divide:

mov eax, ecx

cdq

idiv ebx

mov ecx, eax

jmp get\_operation

modulus:

mov eax, ecx

cdq

idiv ebx

mov ecx, edx

jmp get\_operation

exponentiation:

mov eax, ecx

mov ecx, 1

cmp ebx, 0

je done\_exp

exp\_loop:

imul ecx, eax

dec ebx

jnz exp\_loop

done\_exp:

jmp get\_operation

xorr:

xor ecx, ebx

jmp get\_operation

orr:

or ecx, ebx

jmp get\_operation

andd:

and ecx, ebx

jmp get\_operation

done:

mov eax,red+(black\*16)

call settextcolor

mov edx, offset prompt4

call writestring

mov eax, ecx

call WriteInt

call crlf

call crlf

mov eax,green+(black\*16)

call settextcolor

mov edx, offset prompt3

call writestring

call readint

cmp eax,1

je callagain

cmp eax,0

je ext

ext:

mov eax,cyan+(black\*16)

call settextcolor

mov edx, offset thankyou1

call writestring

mov edx, offset thankyou2

call writestring

mov eax,yellow+(black\*16)

call settextcolor

call crlf

call crlf

exit

main ENDP

END main