section .data ; Data segment

userMsg db 'Please enter a number: ', 0xa, 0xD ; Ask the user to enter a number

lenUserMsg equ $-userMsg ; The length of the message

dispMsg db 'You have entered: ', 0xa, 0xD

lenDispMsg equ $-dispMsg

evenMsg db 'You have entered an even number ', 0xa, 0xD

lenevenMsg equ $-evenMsg

oddMsg db 'You have entered an odd number ', 0xa, 0xD

lenoddMsg equ $-oddMsg

ncMsg db 0xa ; print newline

lenncMsg equ $-ncMsg ; length of newline message

section .bss ; Uninitialized data

num resb 1

num2 resb 1

result resb 1

section .text ; Code Segment

global \_start

\_start:

; User prompt

mov eax, 4

mov ebx, 1

mov ecx, userMsg

mov edx, lenUserMsg

int 80h

; Read and store the user input

mov eax, 3

mov ebx, 0

mov ecx, num

mov edx, 1

int 80h

; Output the message 'You have entered:'

mov eax, 4

mov ebx, 1

mov ecx, dispMsg

mov edx, lenDispMsg

int 80h

; Output the number entered

mov eax, 4

mov ebx, 1

mov ecx, num

mov edx, 1

int 80h

; Print newline carriage return

mov eax, 4

mov ebx, 1

mov ecx, ncMsg

mov edx, lenncMsg

int 80h

; Determine if even or odd

mov al, [num]

sub al, '0'

and al, 1

jz EVEN\_NUMBER

; Have odd number so print that it is odd

mov eax, 4

mov ebx, 1

mov ecx, oddMsg

mov edx, lenoddMsg

int 80h

jmp Continue

EVEN\_NUMBER:

mov eax, 4

mov ebx, 1

mov ecx, evenMsg

mov edx, lenevenMsg

int 80h

Continue:

; Print newline carriage return

mov eax, 4

mov ebx, 1

mov ecx, ncMsg

mov edx, lenncMsg

int 80h

; Read and store another user input number

mov eax, 3

mov ebx, 0

mov ecx, num2

mov edx, 1

int 80h

; Output the message 'The entered number is: '

mov eax, 4

mov ebx, 1

mov ecx, dispMsg

mov edx, lenDispMsg

int 80h

; Output the number entered

mov eax, 4

mov ebx, 1

mov ecx, num2

mov edx, 1

int 80h

; Print newline carriage return

mov eax, 4

mov ebx, 1

mov ecx, ncMsg

mov edx, lenncMsg

int 80h

; Determine if even or odd

mov al, [num2]

sub al, '0'

and al, 1

jz EVEN\_NUMBER2

; Have odd number so print that it is odd

mov eax, 4

mov ebx, 1

mov ecx, oddMsg

mov edx, lenoddMsg

int 80h

jmp exit

EVEN\_NUMBER2:

mov eax, 4

mov ebx, 1

mov ecx, evenMsg

mov edx, lenevenMsg

int 80h

; Exit code

exit:

; Print newline carriage return

mov eax, 4

mov ebx, 1

mov ecx, ncMsg

mov edx, lenncMsg

int 80h

; Exit program

mov eax, 1

int 80h