include irvine32.inc

.data

msg byte "This is Number is Even ",0

msg2 byte "This is odd Number " ,0

.code

main proc

call tell\_even\_or\_not

main endp

tell\_even\_or\_not proc

xor eax,eax

xor ebx ,ebx

mov al,59 ;cheking the number

mov bl,2 ;

div bl

;call dumpregs

cmp ah,0

je yes\_even

jne not\_even

yes\_even:

mov edx,0

mov edx,offset msg

call writestring

call crlf

jmp next

not\_even:

mov edx,0

mov edx,offset msg2

call writestring

call crlf

next:

;ret

tell\_even\_or\_not endp

end main

include irvine32.inc

.data

array byte 4,5,6,7,8

prime\_number byte "this is prime number" ,0

not\_prime\_number byte "This is not prime number ",0

val byte ?

two byte 2

three byte 3

.code

main proc

xor edx,edx

xor ecx,ecx

xor eax,eax

xor ebx,ebx

mov esi,offset array

mov ecx ,lengthof array

L1:

mov al,[esi] ;working fine

mov al, 1 ; Compares eax with 1

JE isPrime ; If eax is == to 1 then it is prime and needs to jump to isPrime

cmp al,2 ; Compares eax with 2

JE isPrime ; If eax is == to 2 then it is prime and needs to jump to isPrime

cmp al,3 ; Compares eax with 3

JE isPrime ; If eax is == to 3 then it is prime and needs to jump to isPrime

mov edx,0

mov val,al ; I store my eax value into val

div two ; I devide by two to see if I get a Remainder in EDX

cmp dl,0 ; I compare edx with 0 in order to see if there is a remainder

JE notPrime ; If this edx is equal to 0 then the number entered is not prime and I skip to notPrime

mov edx,0

mov al,val ; I move the number entered by user back into eax

div three ;because immediate constant are not alolowed here

cmp edx,0 ; I compare edx with 0 in order to see if there is a remainder

JNE isPrime ; I will jump to isPrime if this has a Remainder

;because it meens that it is not 1, 2 or 3 and it is not divisible by 2 and it is not divisable by 3

inc esi

loop L1

isPrime:

mov edx,offset prime\_number

call writestring

call crlf

jmp next

notPrime:

mov edx,offset not\_prime\_number

call writestring

call crlf

next:

main endp

end main