

section .data

msg1 db "entre c/ numero", LF ;10 na tabela ASCII indica o /n veja que foi definido LF como 10

tam equ \$ -msg1 ; faz contagem de caracteres da string acima a ser imprimida

msg2 db "o numero é par", LF

tam2 equ \$ -msg2

msg3 db "o numero é impar", LF

tam3 equ \$ -msg3

LF equ 10 ; fazendo LF equivaler a 10 comando de saltar uma linha

section .bss

numero resb 100

qd resb 100 ;qd recebera o tam da string inserida pelo usuário

section .txt ; nome da diretiva .nome da diretiva

global main

main: ;inicio do programa (gcc)

;orientação ao usuario

mov edx,tam ;Tamanho da string

mov ecx,msg1 ;ponteiro mensagem 1

mov ebx,1 ;Tela (destino) 1 = codigo para saida

mov eax,4 ;Print string (Servico) 4 = codigo para impressao na tela

int 0x80 ;Chamada ao sistema

;carregando a string do teclado

mov edx,100

mov ecx,numero

mov ebx,0

mov eax,3 ;instrução para pegar conteudo do teclado

int 0x80 ;chamando sistema para modificar

mov [qd],eax ;movendo a quantidade de eax para qd que sera o tamanho da string

```

;parte do teste para ver se é par ou impar
sub eax,2

mov esi,eax ;indice so ultimo algarismo

xor edx,edx ;zerando o ponteiro edx

mov al,[numero + esi] ;pegando o ultimo algarismo

mov ebx,2; divisor do algoritmo

div ebx ;divide o eax pelo ebx , eax implicito

    ;o quociente vai para eax

    ;o resto vai para edx

cmp edx,0

je par ;jump para o par

jmp impar

```

```

par: ;parte da msg2 do par

mov edx,tam2 ;Tamanho da string

mov ecx,msg2 ;ponteiro mensagem 1

mov ebx,1 ;Tela (destino) 1 = codigo para saida

mov eax,4 ;Print string (Servico) 4 = codigo para impressao na tela

int 0x80

jmp fim

```

```

impar: ;parte da msg3 do impar

mov edx,tam3 ;Tamanho da string

mov ecx,msg3 ;ponteiro mensagem 1

mov ebx,1 ;Tela (destino) 1 = codigo para saida

mov eax,4 ;Print string (Servico) 4 = codigo para impressao na tela

int 0x80

jmp fim

```

```

fim: ; parte do encerramento

    ;encerra

mov eax,1 ;servico exit

int 0x80 ;chamada ao sistema

```