Palindromo xor 32

```nasm
%macro print 2
mov eax,4
    mov ebx,1
    mov ecx,%1
    mov edx,%2
    int 80h
%endmacro

section .data
        ;cade1 db "abcdefgfedcba"
        cade1 db "AbCDefGfedcbA"
        leng  equ $-cade1
        sip   db "si"
        no    db "no"
section .bss
        buffer: resb 100
section .text
    global _start
_start:
    mov eax,cade1
    mov ebx,(cade1+leng-1)
    mov ecx,(leng/2)
    _for:
        mov byte dl,byte[eax]
        mov byte dh,byte[ebx]
            ;trasnformando a may o min dl,dh
            ;'a' = 97
            ;'A' = 65
            ;hay q restar 32  ; mucho cuidado con eso
            ;por naturaleza pensamos q z es menos q A gggg
            ;ups
            cmp dl,'Z'                ;si minuscula
             jg conti1
             xor dl,32
            conti1:
                        ;salta
            cmp dh,'Z'
             jg conti2
             xor dh,32              ;si es mayus lo combierte a min
    conti2:
        cmp  dh, dl
        jne _no
        inc eax
        dec ebx
    loop _for

_si:
    print sip,2
    mov eax,1
    mov ebx,0
    int 80h
_no:
    print no,2

    mov eax,1
    mov ebx,0
    int 80h
```