

# Dado un archivo mostrar las vocales

```

1 ;1.-Dado un archivo mostrat las vocales
2                                     Listo
3 ;2.-Dado un archivo mostrar las vocales de forma inversa y ver si es
4                                     Espera
5 ; palindromo
6 %macro print_generic 2
7     mov eax,4
8     mov ebx,1
9     mov ecx,%1
10    mov edx,%2
11    int 80h
12 %endmacro
13 section .data
14     tamaño      equ      1000
15     endl        db 10
16     vocales     db "a","e","i","o","u","A","E","I","O","U"
17     space       db " "
18 section .bss
19     buffer:     resb 1000
20 section .text
21     global _start
22 _start:
23     ;extract arguments
24     pop ecx
25     pop ecx
26     pop ecx
27     push ecx
28 print_generic ecx,8
29 print_generic endl,1
30 open_file:
31     mov eax,5          ; open(
32     pop ebx           ; url del archivo
33     mov ecx,0         ; read-only mode
34     int 80h           ; );
35 read_file:
36     mov eax,3          ; read(
37     mov ebx,eax        ; file_descripor,
38     mov ecx,buffer     ; *buf,
39     mov edx,tamaño     ; *bufsize
40     int 80h           ; );
41 print_generic buffer,tamaño
42 print_generic endl,1
43 kernel:
44     mov ecx,buffer
45     push ecx
46     _for:
47         cmp byte[ecx],0          ;end of file or eof
48         je _endfor              ;jump no equal
49         ;compare if is a vocal a,e,i,o,u
50         cmp byte[ecx],'a'
51         jne no_a
52         push ecx
53         print_generic ecx,1
54         pop ecx
55     no_a:
56         cmp byte[ecx],'e'
57         jne no_e

```

```

56         push ecx
57         print_generic ecx,1
58         pop ecx
59     no_e:
60         cmp byte[ecx], 'i'
61         jne no_i
62         push ecx
63         print_generic ecx,1
64         pop ecx
65     no_i:
66         cmp byte[ecx], 'o'
67         jne no_o
68         push ecx
69         print_generic ecx,1
70         pop ecx
71     no_o:
72         cmp byte[ecx], 'u'
73         jne no_vocal
74         push ecx
75         print_generic ecx,1
76         pop ecx
77     no_vocal:
78         push ecx
79         print_generic space,1
80         pop ecx
81         cmp byte[ecx], ' '
82         jne no_space
83
84         push ecx
85         print_generic space,1
86         pop ecx
87     ;end compare vocal
88     no_space:
89         inc ecx                ;index ++
90         jmp _for              ;continue cicle
91 _endfor:
92 fin:
93     mov ebx,0
94     mov eax,1
95     int 80h

```