

```

1  section .bss
2  Buff resb 1          ; hold the value of one char
3
4  section .data
5
6  section .text
7  global _start
8
9  _start:
10 nop                 ; This no-op keeps the debugger happy
11
12 Read:|
13     mov eax,3         ; Specify sys_read call
14     mov ebx,0         ; Specify File Descriptor 0: Standard Input (STDIN)
15     mov ecx,Buff      ; Pass offset of the buffer to read to
16     mov edx,1         ; Tell sys_read to read one char from stdin
17     int 80H           ; Call sys_read
18
19     cmp eax,0         ; Look at sys_read's return value in EAX
20     je Exit           ; Jump If Equal to 0 (0 means EOF) to Exit
21                     ; or fall through to test for lowercase
22
23     cmp byte[Buff],61h ; Test input char against 'a'
24     jae checkLowerCase ; check if the input char is in the lower case range
25
26     cmp byte[Buff],41h ; Test input char against 'A'
27     jae checkUpperCase ; check if the input char is in the upper case range
28
29     jmp Write         ; jump to Write if it is any other signs
30
31 checkUpperCase:
32     cmp byte[Buff],5Ah ; Test input char against 'Z'
33     jbe add13Upper     ; jump to label if cmp is below or equal
34
35 checkLowerCase:
36     cmp byte[Buff],7Ah ; Test input char against 'z'
37     jbe add13Lower     ; jump to label if cmp is below or equal
38
39 add13Lower:
40     cmp byte[Buff],6Eh ; Test input char against 'n'
41     jae sub13          ; jump to label if the input is greater than n
42     add byte[Buff],0Dh  ; adds 13, because the value is less than n
43     jmp Write          ; Write the char to the file
44
45 add13Upper:
46     cmp byte[Buff],4Eh ; Test input char against 'N'
47     jae sub13          ; jump to label if the input is greater than N
48     add byte[Buff],0Dh  ; adds 13, because the value is less than N
49     jmp Write          ; Write the char to the file
50
51 sub13:
52     sub byte[Buff],0Dh  ; subtract 13 because value is higher than the input value
53     jmp Write          ; Write the char to the file
54
55 Write:
56     mov eax,4         ; Specify sys_write call
57     mov ebx,1         ; Specify File Descriptor 1: Standard output (STDOUT)
58     mov ecx,Buff      ; Pass address of the character to write
59     mov edx,1         ; Pass number of chars to write
60     int 80h           ; Call sys_write...
61     jmp Read          ; then go to the beginning to get another char
62
63 Exit:
64     mov eax,1         ; Code for Exit Syscall
65     mov ebx,0         ; Return a code of zero to Linux 80H ; Make kernel call to exit program
66     int 80H           ; Make kernel call to exit program

```

