

## Naming the file: 20BCE1798 ANSH GOEL EX-3 BOOT LOADER PROGRAM

Reg No: 20BCE1798

Name: ANSH GOEL

Course Code: CSE2005

Course Name: Operating Systems (Embedded Lab)

Slot: L27+L28

Ex No 3: Boot\_Loader\_Prg

Date: 28-01-22

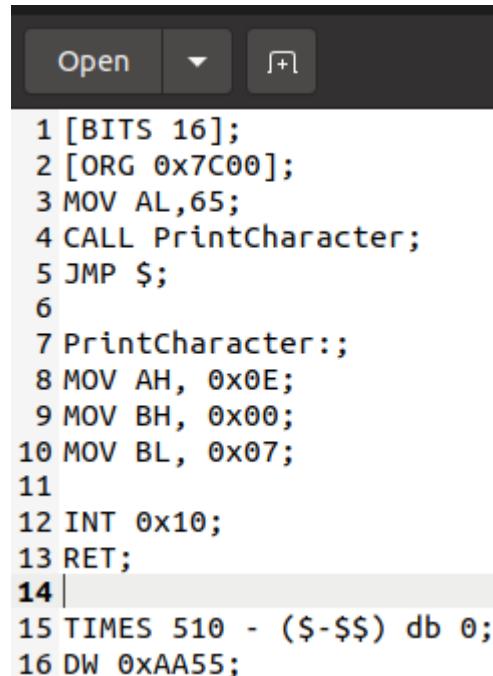
### Ex 3 - Boot Loader

Due 3 February 2022 23:59

#### Instructions

Print A and Hello as 2 different codes with step by step screenshots for one code. Follow upload format.

## CODE: - For printing ‘A’



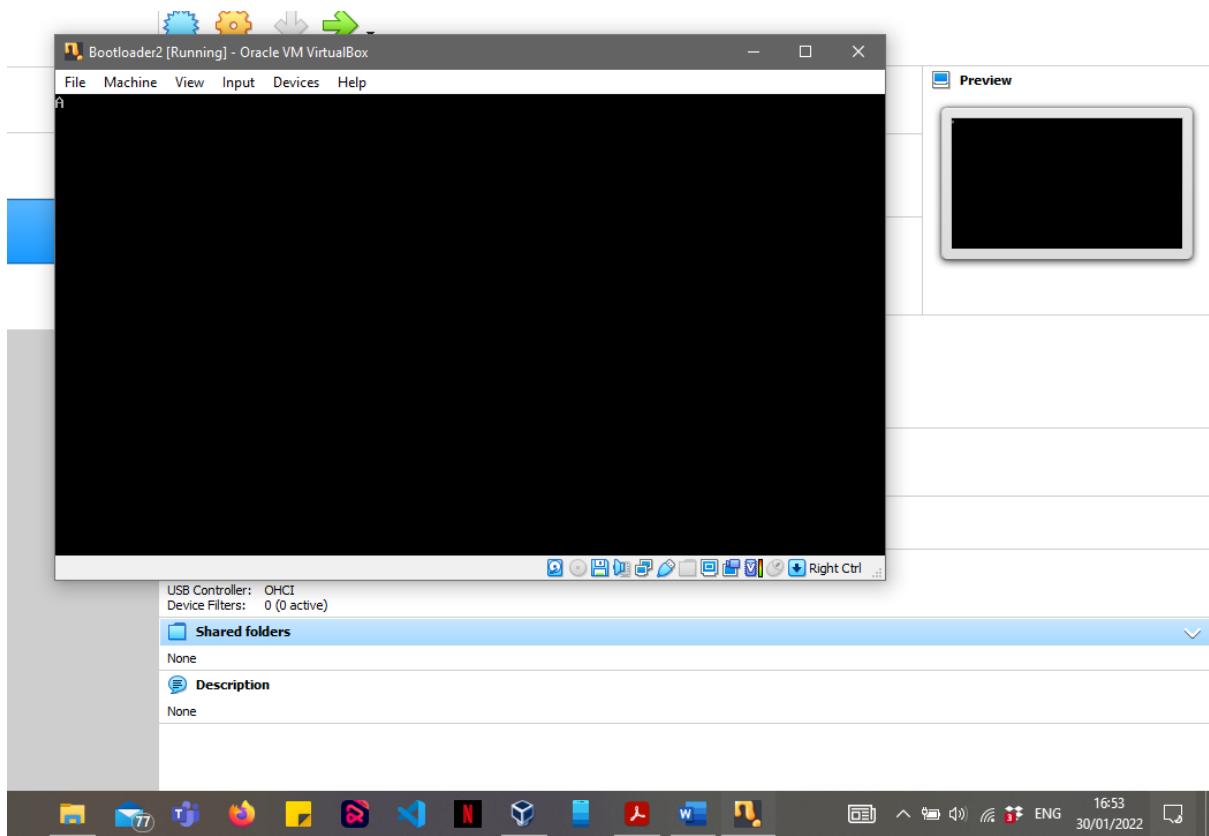
The screenshot shows a hex editor window with a dark theme. At the top, there are three buttons: 'Open' with a dropdown arrow, a '+' button, and a search icon. Below the buttons is a scroll bar. The main area contains assembly code with line numbers on the left:

```
1 [BITS 16];
2 [ORG 0x7C00];
3 MOV AL,65;
4 CALL PrintCharacter;
5 JMP $;
6
7 PrintCharacter:;
8 MOV AH, 0x0E;
9 MOV BH, 0x00;
10 MOV BL, 0x07;
11
12 INT 0x10;
13 RET;
14 |
15 TIMES 510 - ($-$) db 0;
16 DW 0xAA55;
```

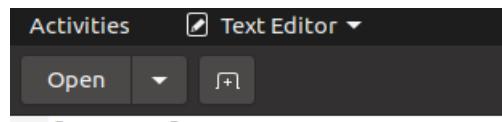
A screenshot of a Windows desktop environment. At the top is a window titled "bootloader2.asm" showing assembly code. The code includes instructions for printing the character 'A' at address 0x7C00, followed by a loop of 510 bytes of zeros. Below this is a standard Windows taskbar with various pinned icons.

```
1 [BITS 16];
2 [ORG 0x7C00];
3 MOV AL,65;
4 CALL PrintCharacter;
5 JMP $;
6
7 PrintCharacter:;
8 MOV AH, 0x0E;
9 MOV BH, 0x00;
10 MOV BL, 0x07;
11
12 INT 0x10;
13 RET;
14|
15 TIMES 510 - ($-$) db 0;
16 DW 0AA55;
```

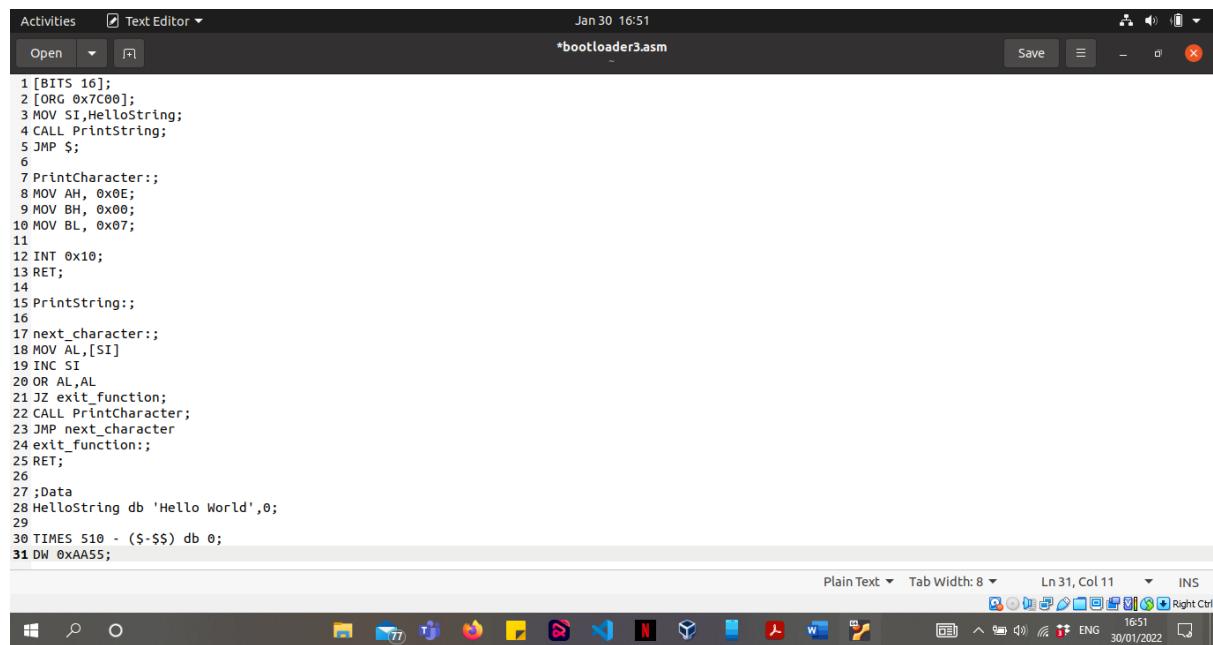
## OUTPUT:



## CODE: For Printing ‘Hello World’



```
1 [BITS 16];
2 [ORG 0x7C00];
3 MOV SI,HelloString;
4 CALL PrintString;
5 JMP $;
6
7 PrintCharacter:;
8 MOV AH, 0x0E;
9 MOV BH, 0x00;
10 MOV BL, 0x07;
11
12 INT 0x10;
13 RET;
14
15 PrintString:;
16
17 next_character:;
18 MOV AL,[SI]
19 INC SI
20 OR AL,AL
21 JZ exit_function;
22 CALL PrintCharacter;
23 JMP next_character
24 exit_function:;
25 RET;
26
27 ;Data
28 HelloString db 'Hello World',0;
29
30 TIMES 510 - ($-$) db 0;
31 DW 0xAA55;
```



```
1 [BITS 16];
2 [ORG 0x7C00];
3 MOV SI,HelloString;
4 CALL PrintString;
5 JMP $;
6
7 PrintCharacter:;
8 MOV AH, 0x0E;
9 MOV BH, 0x00;
10 MOV BL, 0x07;
11
12 INT 0x10;
13 RET;
14
15 PrintString:;
16
17 next_character:;
18 MOV AL,[SI]
19 INC SI
20 OR AL,AL
21 JZ exit_function;
22 CALL PrintCharacter;
23 JMP next_character
24 exit_function:;
25 RET;
26
27 ;Data
28 HelloString db 'Hello World',0;
29
30 TIMES 510 - ($-$) db 0;
31 DW 0xAA55;
```

## OUTPUT:

