



Vidyavardhini's College of Engineering &  
Technology

---

<b>Name:</b>	Viraj Oza
<b>Roll No:</b>	69
<b>Class/Sem:</b>	SE/IV
<b>Experiment No.:</b>	4
<b>Title:</b>	Program to display character in uppercase and lowercase.
<b>Date of Performance:</b>	
<b>Date of Submission:</b>	
<b>Marks:</b>	
<b>Sign of Faculty:</b>	



## Vidyavardhini's College of Engineering & Technology

---

**Aim:** Assembly Language Program to display character A to z in both uppercase and lowercase

**Theory:**

DOS provide various interrupt services that are used by the system programmer. The most commonly used interrupt is INT 21H. It invokes inbuilt DOS functions which can be used to perform various tasks. The most common tasks are reading a user input character from the screen, displaying result on the existing program etc.

In this program, we display the characters A to Z on the DOS prompt. DOS interrupt function 02 displays the contents of DL (ASCII code) on the screen. By loading the ASCII code of 'A' in the DL register, loading AH register with 02h and calling INT 21h it is possible to display character from A to Z on the screen.

INT 21h/AH = 2 - write character to standard output.

Entry: DL = character to write, after execution AL = DL.

**Example :-**

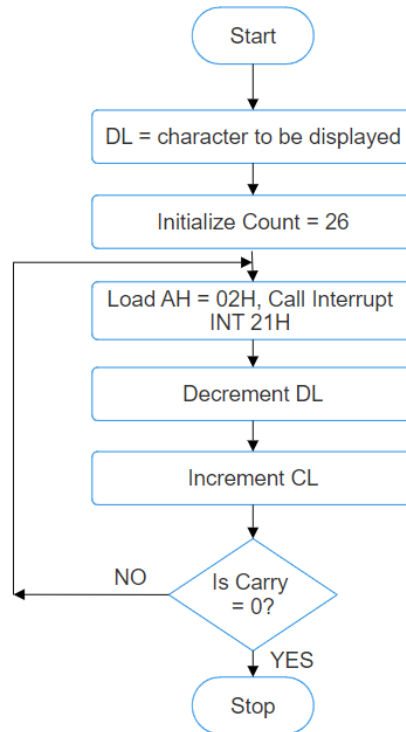
```
mov ah , 2
```

```
mov dl , 'a'
```

```
int 21h
```



## Flowchart:



## Algorithm:

1. Start.
2. Initialize DL with 'A'.
3. Load CL with count = 26.
4. Load AH = 02H and call INT 21H.
5. Increment DL, to next character.
6. Decrement the count.
7. Repeat steps 4,5,6 till CL is not zero.
8. To end the program use DOS interrupt:
  - 1) Load AH = 41H.
  - 2) Call INT 21 H.
9. Stop.



Program:

org 100h

mov cx, 1Ah

mov dl,'a'

L1:

mov ah, 02h

int 21h

inc dl

dec cx

jnz L1

mov dl, 0ah

int 21h

mov dl, 0dh

int 21h

mov cx, 26

mov dl,'A'

L2:

mov ah, 02h

int 21h

inc dl

dec cx

jnz L2

ret



The screenshot displays the x86-64 emulator interface. At the top, a title bar reads "emulator screen (80x25 chars)". Below it is a terminal window showing the ASCII alphabet in both lowercase and uppercase. The main interface includes a menu bar with options: file, math, debug, view, external, virtual devices, virtual drive, help. Below the menu is a toolbar with icons for Load, reload, step back, single step, run, and a step delay slider set to 0 ms. The central area is divided into three main sections: registers, memory, and assembly code.

**Registers Section:**

	H	L
AX	02	5A
BX	00	00
CX	00	00
DX	00	5B
CS	F400	
IP	0154	
SS	0700	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

**Memory Section:**

Address range: F400:0150 - F400:0154

Address	Hex	Dec	Comment
F4150:	FF	255	RES
F4151:	FF	255	RES
F4152:	CD	205	=
F4153:	20	032	SPA
F4154:	CF	207	±
F4155:	00	000	NULL
F4156:	00	000	NULL
F4157:	00	000	NULL
F4158:	00	000	NULL
F4159:	00	000	NULL
F415A:	00	000	NULL
F415B:	00	000	NULL
F415C:	00	000	NULL
F415D:	00	000	NULL
F415E:	00	000	NULL
F415F:	00	000	NULL
F4160:	FF	255	RES
F4161:	FF	255	RES
F4162:	CD	205	=
F4163:	1A	026	→
F4164:	CF	207	±
F4165:	00	000	NULL
F4166:	00	000	NULL
F4167:	00	000	NULL
F4168:	00	000	NULL
F4169:	00	000	NULL

**Assembly Code Section:**

```

10 dec cx
11 jnz L1
12
13 mov dl, 0ah
14 int 21h
15 mov dl, 0dh
16 int 21h
17
18 mov cx, 26
19 mov dl, 'A'
20 L2:
21 mov ah, 02h
22 int 21h
23 inc dl
24 dec cx
25 jnz L2
26
27 ret
  
```

At the bottom of the interface is a status bar with buttons for screen, source, reset, aux, vars, debug, stack, and flags.

1. Explain INT 21H.

Here are some of the commonly used functions provided by `INT 21h`:

- AH = 0x3D: Open File
- AH = 0x3E: Close File
- AH = 0x3F: Read from File
- AH = 0x40: Write to File
- AH = 0x42: Move File Pointer
- AH = 0x4C: Terminate Program with Return Code

- AH = 0x02: Display Character



## Vidyavardhini's College of Engineering & Technology

---

- AH = 0x09: Display String

### Keyboard Input/Output:

- AH = 0x01: Read Character from Standard Input
- AH = 0x06: Direct Console Output
- AH = 0x07: Direct Console Input
- AH = 0x08: Read Character with Echo

### Miscellaneous:

- AH = 0x00: Terminate Program
- AH = 0x4A: Resize Memory Block

When using `INT 21h`, the specific function to be performed is specified in the AH register, and additional parameters or data may be passed in other registers or memory locations, depending on the function being called.

For example, to display a character on the screen using `INT 21h`, you would load the ASCII value of the character into the AL register and set AH to 0x02. Then you would trigger the interrupt using the `INT 21h` instruction.

### 2. Explain working of increment and decrement instructions.

Ans. Increment Instruction: Syntax: `INC destination`. Function: The increment instruction increases the value of the destination operand by one. The destination operand can be a register or a memory location. After the increment operation, the Zero Flag (ZF) and Sign Flag (SF) in the CPU's flags register are updated based on the result. If the result is zero, the ZF is set; if the result is negative, the SF is set. Example: `INC AX` increases the value in the AX register by one.

Decrement Instruction: Syntax: `DEC destination`. Function: The decrement instruction decreases the value of the destination operand by one. Similar to the increment instruction, the destination operand can be a register or a memory location. After the decrement operation, the Zero Flag (ZF) and Sign Flag (SF) in the CPU's flags register are updated based on the result. If the result is zero, the ZF is set; if the result is negative, the SF is set. Example: `DEC BX` decreases the value in the BX register by one.