



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>	5
<b>Title:</b>	Program to display string in Lowercase.
<b>Date of Performance:</b>	
<b>Date of Submission:</b>	
<b>Marks:</b>	
<b>Sign of Faculty:</b>	



## Vidyavardhini's College of Engineering & Department of Artificial Intelligence and Data Science (AI&DS)

---

**Aim:** Program to display string in Lowercase.

**Theory:**

The program will take Uppercase string as input and convert it to lowercase string. Int 21h is a DOS interrupt. To use the DOS interrupt 21h load with the desired sub-function. Load other required parameters in other registers and make a call to INT 21h.

INT 21h/AH = 9

output of string at DS: • String must be terminated by "\$"

example :

```
org 100h
```

```
mov dx, offset msg
```

```
mov ah, 9
```

```
int 21h
```

```
ret
```

```
msg db "hello world $"
```

INT 21h/AH = 0AH – input of string to DS:DX, first byte is buffer size, second byte is number of chars actually read this function does not add '\$' in the end of string to print using INT 21h/AH = 9 you must set dollar character at the end of it and start printing from address DS : DX + 2. The function does not allow to enter more characters than the specified buffer size.



**Algorithm:**

1. Start.
2. Initialize the Data Segment.
3. Display message -1.
4. Input the string.
5. Display message-2.
- 6 Take the character count in CX.
7. Point to the first character.
8. Convert it to Lowercase.
9. Display the character.
10. Decrement the character coun.
11. If not Zero, repeat from step 6.
12. To terminate the program, using the DOS interrupt:
  - 1) Initialize AH with 4CH
  - 2) Call interrupt INT 21H.
13. Stop.



Program:

```
org 100h
```

```
.data
```

```
m1 db 10,13,'Enter the string in uppercase:$'
```

```
m2 db 10,13,'The lowercase string is:$'
```

```
buff db 80
```

```
.code
```

```
lea dx,m1
```

```
mov ah,09h
```

```
int 21h
```

```
lea dx,buff
```

```
mov ah,0ah
```

```
int 21h
```

```
lea dx,m2
```

```
mov ah,09h
```

```
int 21h
```

```
mov cl,[buff+1]
```

```
lea bx,buff+2
```

```
l1:
```

```
mov dx,[bx]
```

```
add dx,20h
```

```
mov ah,02h
```

```
int 21h
```

```
inc bx
```

```
loop l1
```

```
ret
```



Output:

The screenshot shows an x86 emulator window titled "emulator screen (80x25 chars)". The text inside the window reads: "Enter the string in uppercase:YASH" and "The lowercase string is:yash". Below the text is a menu bar with options: file, math, debug, view, external, virtual devices, virtual drive, help. There are buttons for Load, reload, step back, single step, run, and a step delay slider set to 0 ms. On the left, a registers window shows the state of various registers. The IP register is highlighted at 0154. The main window displays memory contents starting from F4150:0154. The memory contains the string "YASH" in uppercase. On the right, a window titled "original source co..." shows the assembly code used for the conversion. The code includes instructions like "lea dx,m2", "mov ah,09h", "int 21h", "mov cl,[buff+1]", "lea bx,buff+2", "l1:", "mov dx,[bx]", "add dx,20h", "mov ah,02h", "int 21h", "inc bx", "loop l1", and "ret".

**Conclusion:** We successfully converts the input string to lowercase characters, providing a simple and efficient method for transforming text to lowercase format.

## 1. Explain instruction AAA.

Ans. The instruction "AAA" stands for "ASCII Adjust After Addition." It is a mnemonic used in x86 assembly language programming. This instruction adjusts the result of an addition operation in the AL (accumulator) register to ensure that it remains in the ASCII character range (0x30 to 0x39 for digits '0' to '9'). If the AL register contains a value greater than 9 or the Auxiliary Carry Flag (AF) is set, it adjusts the AL register by adding 6 to it and increments the AH (high byte of AX) register by 1. This instruction is commonly used when performing arithmetic operations on ASCII-coded decimal values to ensure proper representation.

## 2. Explain instruction AAS.

Ans. The instruction "AAS" stands for "ASCII Adjust After Subtraction." It is a mnemonic used in x86 assembly language programming. This instruction adjusts the result of a subtraction operation in the AL (accumulator) register to ensure that it remains in the ASCII character range (0x30 to 0x39 for digits '0' to '9'). If the AL register contains a value greater than 9 or the Auxiliary Carry Flag (AF) is set, it adjusts the AL register by subtracting 6 from it and decrements the AH (high byte of AX) register by 1. This instruction is commonly used when performing arithmetic operations on ASCII-coded decimal values to ensure proper representation.



Vidyavardhini's College of Engineering & Department  
of Artificial Intelligence and Data Science (AI&DS)

---