Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2024-25

Class:	**SE**	**Semester:**	**IV**
Course Code:	**CSL404**	**Course Name:**	**Microprocessor Lab**

Name of Student:	**Bhagyashri Kaleni Sutar**
Roll No. :	**75**
Experiment No.:	**4**
Title of the Experiment:	**Program to display alphabets A to Z in uppercase and lowercase**
Date of Performance:	**03/02/2026**
Date of Submission:	**10/02/2025**

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Ms. Sweety Patil

Signature :

Date:

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 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.

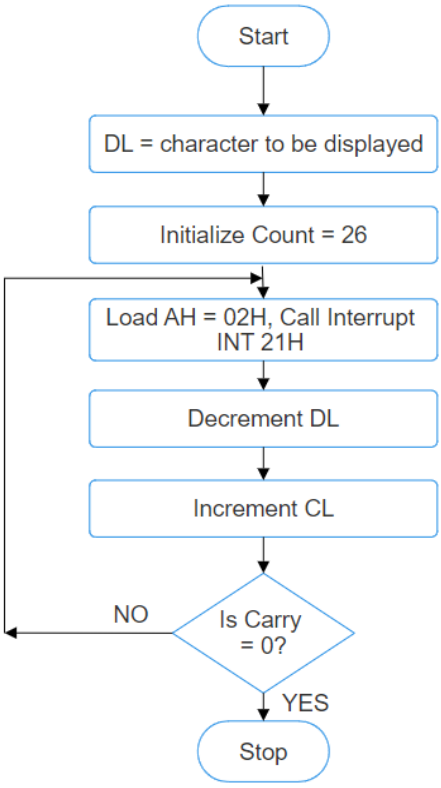
Example :-

```
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.

Code:

original source code

```
01 mov cx,26
02 mov dl,'A'
03 L1:mov ah,02h
04 int 21h
05 inc dl
06 dec cx
07 jnz L1
08 mov dl, 0Dh
09 mov ah, 02h
10 int 21h
11 mov dl, 0Ah
12 int 21h
13 mov cx,26
14 mov dl,'a'
15
16 L2:mov ah,02h
17 int 21h
18 inc dl
19 dec cx
20 jnz L2
21
22
23
```

Output:

Conclusion:

1. Explain INT 21H.

Ans:- **INT 21h** is a DOS (Disk Operating System) interrupt that provides a wide range of services for interacting with the operating system. It serves as the primary mechanism for system calls in DOS, allowing software programs to perform various tasks such as file management, input/output operations, memory management, and more. Each function is accessed by setting specific values in the CPU registers, particularly in the AH register.

Key Features of INT 21h:

- **System-level Interface**: It provides a way for programs to communicate with the underlying operating system to perform actions that the hardware cannot directly handle.
- **Wide Range of Functions**: INT 21h encompasses a wide variety of functions for managing files, input/output, memory, and program termination, among other tasks.

2. Explain working of increment and decrement instructions.

Ans:- In assembly language programming, **increment** and **decrement** are common operations that modify the value of a register or memory location by 1. These operations are widely used in loops, counters, and algorithms. The instructions for increment and decrement are simple, efficient, and essential for managing the flow of a program.

1. Increment Instruction (INC):

The **increment** instruction increases the value of a specified operand (register or memory location) by 1. It is used to add one to a value without affecting other flags except the Zero Flag (ZF) and Sign Flag (SF).

Syntax:- INC operand

- **operand**: This can be a register, a memory location, or a memory address.
- It increments the value of the operand by 1.

Ex:-

```
mov ax, 5     ; AX register is set to 5
inc ax        ; Increment AX by 1 (AX becomes 6)
```

2. Decrement Instruction (DEC):

The **decrement** instruction decreases the value of a specified operand (register or memory location) by 1. It is used to subtract one from a value and also affects certain flags.

Syntax:

DEC operand

- **operand**: This can be a register, a memory location, or a memory address.
- It decrements the value of the operand by 1.