1. **.MODEL SMALL**
   * Specifies the "small" memory model, where code and data fit within separate 64 KB segments.
2. **.STACK 100H**
   * Allocates 256 bytes (100H) for the stack.
3. **.DATA**
   * Declares the **data segment**, which stores program constants and variables.
     + PROMPT\_1 DB 'Enter a Lowercase letter : $': String to prompt the user for input. The $ marks the end of the string.
     + PROMPT\_2 DB 0AH,0DH,'Upper case : $': String to label the output. 0AH and 0DH are ASCII codes for newline and carriage return, ensuring the output appears on a new line.
     + CHAR DB '?''$': A variable to store the uppercase character. Initially set to '?'.
4. **.CODE**
   * Begins the **code segment**, where the logic resides.

**Main Procedure**

**Step-by-Step Breakdown**

1. **Initialize the Data Segment**

assembly

Copy code

MOV AX, @DATA

MOV DS, AX

* + Loads the address of the DATA segment into the AX register and assigns it to DS (Data Segment register). This initializes the DATA segment for use.

1. **Display the Input Prompt**

assembly

Copy code

**MOV AH, 9**

**LEA DX, PROMPT\_1**

**INT 21H**

* + MOV AH, 9: DOS interrupt function to display a string.
  + LEA DX, PROMPT\_1: Loads the address of PROMPT\_1 into DX.
  + INT 21H: Displays the string:

css

Copy code

Enter a Lowercase letter :

1. **Read the User's Input**

assembly

Copy code

**MOV AH, 1**

INT 21H

* + MOV AH, 1: DOS interrupt function to read a single character input.
  + INT 21H: The character entered by the user is stored in AL (the lower 8 bits of the AX register).

1. **Convert the Lowercase Letter to Uppercase**

assembly

Copy code

**SUB AL, 20H**

**MOV CHAR, AL**

* + SUB AL, 20H: Converts the lowercase ASCII character to uppercase by subtracting 20H (hexadecimal for 32). This works because in ASCII, the uppercase letters are exactly 32 positions lower than their lowercase counterparts.
    - Example:
      * a (97) - 32 = A (65)
      * b (98) - 32 = B (66)
  + MOV CHAR, AL: Stores the resulting uppercase letter in the variable CHAR.

1. **Display the Output Label**

assembly

Copy code

MOV AH, 9

LEA DX, PROMPT\_2

INT 21H

* + MOV AH, 9: Prepares DOS to display a string.
  + LEA DX, PROMPT\_2: Loads the address of PROMPT\_2 into DX.
  + INT 21H: Displays the string:

arduino

Copy code

Upper case :

1. **Display the Uppercase Character**

assembly

Copy code

MOV AH, 2

MOV DL, CHAR

INT 21H

* + MOV AH, 2: DOS interrupt function to display a single character.
  + MOV DL, CHAR: Loads the uppercase character stored in CHAR into DL (the lower 8 bits of the DX register).
  + INT 21H: Displays the uppercase character.

1. **Exit the Program**

assembly

Copy code

MOV AH, 4CH

INT 21H

* + MOV AH, 4CH: DOS interrupt function to terminate the program.
  + INT 21H: Ends the program and returns control to the operating system.

**Input and Output Example**

**Input**

* User enters the lowercase letter: a

**Output**

The program will display:

less

Copy code

Enter a Lowercase letter :

Upper case : A