**segemnt**

1. **.MODEL SMALL**
   * Specifies the memory model. The "small" model is used, which means the code and data segments are each limited to 64 KB.
2. **.STACK 100H**
   * Allocates a 256-byte stack (100H in hexadecimal). This is required for temporary storage during execution.
3. **.DATA**
   * This segment is used to define program data.
   * MSG DB 'print from A to Z : $':
     + Defines a string MSG that will be displayed on the screen. The $ symbol is a termination character for string printing in DOS.
4. **.CODE**
   * Begins the code segment where the instructions are written.

**Main Procedure**

assembly

Copy code

MAIN PROC

This starts the main procedure where the actual code logic resides.

**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 (@DATA) into the AX register.
  + Transfers the address in AX to the DS register, initializing the data segment.

1. **Print the message**

assembly

Copy code

MOV AH, 9

LEA DX, MSG

INT 21H

* + MOV AH, 9: Sets the DOS interrupt function to print a string.
  + LEA DX, MSG: Loads the address of the string MSG into DX.
  + INT 21H: Triggers the DOS interrupt to print the string MSG.

Output so far:

css

Copy code

print from A to Z :

1. **Setup for printing characters (A to Z)**

assembly

Copy code

MOV CX, 26

MOV AH, 2

MOV DL, 65

* + MOV CX, 26: Initializes the loop counter (CX) to 26 (number of letters in the alphabet).
  + MOV AH, 2: Prepares DOS function 2 for printing a single character.
  + MOV DL, 65: Initializes DL with 65, the ASCII value of A.

1. **Print loop**

assembly

Copy code

@LOOP:

INT 21H

INC DL

DEC CX

JNZ @LOOP

* + @LOOP:: Marks the start of the loop.
  + INT 21H: Calls DOS to print the character in DL.
  + INC DL: Increments DL to move to the next ASCII character.
  + DEC CX: Decrements the loop counter.
  + JNZ @LOOP: If CX is not zero, the loop continues.

This loop prints characters from A (ASCII 65) to Z (ASCII 90).

1. **Exit the program**

assembly

Copy code

MOV AH, 4CH

INT 21H

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