Structure of an Assembly Language Program for 8086 Microprocessor

```
.MODEL memory_model
.STACK stack_size

.DATA
; programs data goes here

.CODE

MAIN PROC
; main procedure goes here
MAIN ENDP

; Sub procedures goes here
END MAIN
```

Description of the Program Structure

Here,

- .MODEL is mandatory to declare different memory sizes needed for program and it is generally SMALL size.

- .STACK (stack segment) and .CODE (code segment) are mandatory. Although in a program there is no use of stack but a size of 100H (256 bytes) stack declaration is mandatory for internal use of the program. On the other hand, without code segment no program is created.

- DATA segment is optional and necessary only when there is some data used in program.

- In CODE segment there must be only one MAIN procedure and there may be some sub procedures (subprograms or subroutines or user-defined functions)

- To indicate a comment semicolon (;) is used.

An Example

Problem-1 : Write an assemble language program that will display a character.

Solution :

```
.MODEL SMALL
.STACK 100H

.CODE
MAIN PROC
MOV DL,'A'
MOV AH,02H
INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

Description of the Program

For Single Character Output :

MOV DL,'A'
MOV AH,02H
INT 21H

1. **The output character or it's ASCII value (in decimal or hex form) must be loaded in DL register.**
2. **The AH register is loaded with function number 2 (for single character output).**
3. **Interrupt 33 or 21H (INT 21H) is invoked for indicating a DOS interrupt for an I/O operation.**

Here,

MOV AH,4CH
INT 21H

- Function number 76 (4CH) and Interrupt 33 (21H) are used for exit to DOS.

Input-Output Operation with Function Number, Interrupt Number and Associated Registers

Operation	Category	Through Reg.	Function No.	Function Reg.	Interrupt No.
Input	Single Character	AL	01H	AH	21H
Output	Single Character	DL	02H	AH	21H
Output	String (offset)	DX	09H	AH	21H

How to Edit, Save, Assemble, Link and Run an Assembly Language Program using 8086 Microprocessor

Edit :

- To create or edit any assembly language program an ASCII text editor such as Notepad program is used.

Save :

- To save any assembly language program created in an ASCII text editor a file name with .asm (assembly) extension is used. Example: problem1.asm

- **Now go to command prompt to give DOS commands for Assemble, Link, and Run the assembly language program.**

- To Assemble, Link or Run any assembly language program, suppose, the assembler program (MASM.exe) is available with a simulator program MASM 6.11 or MASM 6.15. Here, both the source program (problem1.asm) and assembler program (MASM.exe) are kept in same directory (suppose d:\MASM611\BIN) before assemble, link, or run the source program.

Assemble :

The command to assemble any assembly language program is below:

D:\MASM611\BIN\masm filename;

Link :

The command to link any assembly language program is below:

D:\MASM611\BIN\link filename;

Run :

The command to run any assembly language program is below:

D:\MASM611\BIN\filename;

Problem-2 : Write an assemble language program that will take a character as input and then display the inputted character.

Solution :

```
.MODEL SMALL
.STACK 100H

.CODE
MAIN PROC
MOV DL,'?'
MOV AH,02H
INT 21H
MOV AH,01H
INT 21H
MOV DL,AL
MOV AH,02H
INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

Use of Data Segment

Problem-3 : Write an assembly language program that will display the message ‘Bangladesh is our country’.

```
.MODEL SMALL
.STACK 100H

.DATA
MSG DB ‘Bangladesh is our country$’
.CODE
MAIN PROC
MOV AX, @DATA
MOV DS,AX
LEA DX,MSG
MOV AH,09H
INT 21H
MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```

Program Description

Here,

- MSG is a variable and it's type is Byte (DB-Define Byte).
- The dollar sign ($) indicates the termination of a string.
- MOV AX, @DATA and MOV DS,AX are used to initialize Data Segment register (DS). No direct initialization is possible to segment registers.
- LEA means Load Effective Address (offset address or starting address) of the string.