Sub :

Day

Time : Date : / /

Md. Nur-E-Azam
ID : 151226802
Section : 07
CSE - 331
LAB - 3

In this Assembly Language programming, A single program is divided into four segments which are-

1. Data Segment,
2. Code Segment,
3. Stack Segment, and
4. Extra Segment.

Print : Hello World in Assembly Language

```
DATA SEGMENT
    MESSAGE DB "HELLO WORLD!!!$"
ENDS
CODE SEGMENT
    ASSUME DS: DATA CS: CODE
START:
    MOV AX, DATA
    MOV DS, AX
```

Sub : Day Time : Date : / /

```
    LEA DX, MESSAGE
    MOV AH, 9
    INT 21H
    MOV AH, 4CH
    INT 21H
ENDS
END START
```

□ Assembly Example 1 — Print 2 Strings

```
.MODEL SMALL
.STACK 100H
.DATA
    STRING_1 DB 'I hate CSE331 $'
    STRING_2 DB 'But I Love Kacchi!!! $'
.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    LEA DX, STRING_1
    MOV AH, 9
    INT 21H
```

Sub : _____ Day | | | | | | | | Time : Date : / /

```
MOV AH,2
MOV DL, ODH
INT 21H
MOV DL, OAH
INT 21H

LEA DX, STRING_2
MOV AH, 9
INT 21H

MOV AH, 4CH
INT 21H

MAIN ENDP
END MAIN
```

☐ Assembly Example 2 – Read a string and Print it

```
• MODEL SMALL
• STACK 100H
• DATA
    MSG_1 EQU "Enter the character : $"
    MSG_2 EQU ODH, OAH, "The given character is: $"
    PROMPT_1 DB MSG_1
    PROMPT_2 DB MSG_2
```

Sub :

Day

Time : Date : / /

- CODE

```
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    LEA DX, PROMPT_1
    MOV AH, 9
    INT 21H

    MOV AH, 1
    INT 21H

    MOV BL, AL

    LEA DX, PROMPT_2
    MOV AH, 9
    INT 21H

    MOV AH, 2
    MOV DL, BL
    INT 21H

    MOV AH, 4CH
    INT 21H
MAIN ENDP
END MAIN
```

Sub :

Day

Time : Date : / /

□ Assembly Example 3 - Read a string from user and display this string in a new line.

```
• MODEL SMALL
• STACK 100H

• CODE
  MAIN PROC
        mov AH, 1
        INT 21H

        mov BL, AL

        mov AH, 2
        mov DL, 0DH
        INT 21H

        MOV DL, 0AH
        INT 21H

        mov AH, 2
        mov DL, BL
        INT 21H

        mov AH, 4CH
        INT 21H
      MAIN ENDP
END MAIN
```

Sub :

Day

Time : Date : / /

□ Assembly Example 4 - Read a string with gaps and print it.

```
.MODEL SMALL
.STACK 64
.DATA

    STRING DB ?
    SYM DB '$'
    INPUT_M DB 0ah, 0dh, 0AH, 0DH, "Enter the
    Input', 0DH, 0AH, '$'

    Outpnt_M DB 0ah, 0dh, 0AH, 0DH,
    'The output is", 0DH, 0AH, '$'
.CODE
MAIN PROC
        MOV AX, @DATA
        MOV DS, AX
        MOV DX, OFFSET INPUT_M
        MOV AH, 09
        INT 21H
        LEA SI, STRING

INPUT:  MOV AH, 01
        INT 21H
```

Sub : _____ Day | Time : Date : / /

```
        mov [SI], AL

        INC SI
        cmp AL, 0DH
        JNZ INPUT
        MOV [SI], '$'

OUTPUT : LEA DX, OUTPUT_M
         MOV AH, 9
         INT 21H

         mov DL, 0AH
         MOV AH, 02H
         INT 21H
         mov DX, OFFSET STRING
         MOV AH, 09H
         INT 21H
         MOV AH, 4CH
         INT 21H

MAIN ENDP

END MAIN
```

Sub :

Day

Time : Date : / /

Assembly Example 5 - printing string using MOV instr

```
.model small
.STACK 100H
.DATA
    MSG1 DB 'KIT!! Lemon Lage :D $'
.CODE
    mov AX, @DATA
    mov DS, AX
    mov dx, OFFSET MSG1
    mov ah, 09h
    int 21h
    mov ah, 4ch
    int 21h
END
```

Assembly Example 6 - print Digit from 0-9

```
.MODEL SMALL
.STACK 100H
.DATA
    PROMPT DB 'The counting from 0 to 9 is
.CODE
    MAIN PROC
    MOV AX, @DATA
```

Sub : _____ Day | Time : Date : / /

```
mov  ds, ax
LEA  dx, prompt
mov  ah, 9
INT  21h
mov  cx, 10
mov  ah, 2
mov  dl, 48
@Loop:
    INT 21h
    INC dl
    DEC cx
    JNZ @Loop
    Mov ah, 4ch
    INT 21h
    MAIN ENDP
END MAIN
```

Assembly Example 7 - Sum of two integers

```
.MODE SMALL
.STACK 100h
.DATA
    PROMT-1 DB 'Enter the First digit : $'
```

Sub :

Day

Time : Date : / /

```
PROMPT_2    DB 'Enter the Second digit
PROMPT_3    DB 'sum of First and second
VALUE_1     DB ?
VALUE_2     DB ?

.CODE
  MAIN PROC
      MOV AX, @DATA
      MOV DS, AX

      LEA DX, PROMPT_1
      MOV AH, 9
      INT 21H

      MOV AH, 1
      INT 21H

      SUB AL, 30H
      MOV VALUE_1, AL

      MOV AH, 2
      MOV DL, 0DH
      INT 21H
```

Sub :

Day

Time : Date : / /

```
LEA   DX, PROMPT_2
MOV   AH, 9
INT   21H

MOV   AH, 1
INT   21H
SUB   AL, 30H
MOV   VALUE_2, AL

MOV   AH, 2
MOV   DL 0DH
INT   21H

MOV   DL, 0AH
INT   21H

LEA   DX, PROMPT, 3
MOV   AH, 9
INT   21H

MOV   AL, VALUE_1
ADD   AL, VALUE_2
ADD   AL, 30H
MOV   AH, 2
MOV   DL, AL
INT   21H

MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN
```