Lab 2 Sadia Tasnim 19101526 Section 06 CSE341

Task 01

Take two numbers as input, add them, print the message "The **result is**" and then display the result in the next line.

Sample Execution

3 (1st input)
1 (2nd input)
The result is
4

edit: F:\9TH\341 LAB\LabA2_Task01.asm

e edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
;Task 01
;Take two numbers as input, add them, print the message "The result is" and then
;display the result in the next line.

.MODEL SMALL

.STACK 100H

.DATA
    ; DEFINE YOUR VARIABLES HERE
    S DB "The result is $"
.CODE
    MAIN PROC

        MOV AX, @DATA
        MOV DS, AX

        ; YOUR CODE STARTS HERE


        MOV AH,1    ;string char input for num 1
        INT 21H     ;interrupt
        MOV BL,AL   ;BL=AL=num 1 input

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        MOV AH,1    ;string char input for num 2
        INT 21H     ;interrupt

        ADD BL,AL   ;BL = BL+AL = num1+ num2

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        LEA DX,S    ;load effective address
        MOV AH,9    ;string output
        INT 21H     ;interrupt

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        MOV DL,BL   ;DL = BL = num1 + num2
        SUB DL,30H  ;DL=DL-30H= num1+num2-30H

        MOV AH,2    ;string char output
        INT 21H     ;interrupt

        ; YOUR CODE ENDS HERE

        MOV AX, 4C00H
        INT 21H

    MAIN ENDP
    END MAIN
```

```
;Task 01
;Take two numbers as input, add them, print the message "The result is" and then
;display the result in the next line.


.MODEL SMALL

.STACK 100H

.DATA
    ; DEFINE YOUR VARIABLES HERE
    S DB "The result is $"
.CODE
    MAIN PROC

        MOV AX, @DATA
        MOV DS, AX

        ; YOUR CODE STARTS HERE


        MOV AH,1   ;string char input for num 1
        INT 21H    ;interrupt
        MOV BL,AL  ;BL=AL=num 1 input

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H    ;interrupt
        MOV DL, 0AH;line feed
        INT 21H    ;interrupt

        MOV AH,1   ;string char input for num 2
        INT 21H    ;interrupt

        ADD BL,AL  ;BL = BL+AL = num1+ num2

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H    ;interrupt
        MOV DL, 0AH;line feed
        INT 21H    ;interrupt

        LEA DX,S   ;load effective address
        MOV AH,9   ;string output
        INT 21H    ;interrupt
```

```
    MOV AH, 2
    MOV DL, 0DH;carriage return
    INT 21H    ;interrupt
    MOV DL, 0AH;line feed
    INT 21H    ;interrupt

    MOV DL,BL  ;DL = BL = num1 + num2
    SUB DL,30H ;DL=DL-30H= num1+num2-30H

    MOV AH,2   ;string char output
    INT 21H    ;interrupt

    ; YOUR CODE ENDS HERE

    MOV AX, 4C00H
    INT 21H

MAIN ENDP
END MAIN
```

Task 02

Take a lowercase letter as input from the user (**omit error checking**, so you do not have to write code for checking whether the user is inputting upper case or lower case), and display it at the next **position** on the **next line** in **upper case**.

Sample Execution 1

a

A

Sample Execution 2

b

B

edit: F:\9TH\341 LAB\LabA2_Task02.asm

edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
;Task 02
;Take a lowercase letter as input from the user (omit error checking, so you do not
;have to write code for checking whether the user is inputting upper case or lower
;case), and display it at the next position on the next line in upper case.

.MODEL SMALL

.STACK 100H

.DATA
    ; DEFINE YOUR VARIABLES HERE
.CODE
    MAIN PROC

        MOV AX, @DATA
        MOV DS, AX

        ; YOUR CODE STARTS HERE


        MOV AH,1    ;string char input for lowercase letter
        INT 21H     ;interrupt
        MOV DL,AL   ;DL=lowercase letter ascii value

        SUB DL,20H ;DL=DL-20H =upperrcase letter ascii value
        MOV BL,DL   ;BL=DL =upperrcase letter ascii value

        MOV AH,2
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        MOV DL,BL   ;DL=BL =upperrcase letter ascii value
        MOV AH,2    ;string char output
        INT 21H     ;interrupt


        ; YOUR CODE ENDS HERE

        MOV AX, 4C00H
        INT 21H

    MAIN ENDP
```

```
;Task 02
;Take a lowercase letter as input from the user (omit error checking, so you do not
;have to write code for checking whether the user is inputting uppercase or lower
;case), and display it at the next position on the next line in upper case.

.MODEL SMALL

.STACK 100H

.DATA
   ; DEFINE YOUR VARIABLES HERE
.CODE
   MAIN PROC

      MOV AX, @DATA
      MOV DS, AX

      ; YOUR CODE STARTS HERE

      MOV AH,1   ;string char input for lowercase letter
      INT 21H    ;interrupt
      MOV DL,AL  ;DL=lowercase letter ascii value

      SUB DL,20H ;DL=DL-20H =uppercase letter ascii value
      MOV BL,DL  ;BL=DL =uppercase letter ascii value

      MOV AH,2
      MOV DL, 0AH;line feed
      INT 21H    ;interrupt

      MOV DL,BL  ;DL=BL =uppercase letter ascii value
      MOV AH,2   ;string char output
      INT 21H    ;interrupt


      ; YOUR CODE ENDS HERE

      MOV AX, 4C00H
      INT 21H

   MAIN ENDP
   END MAIN
```

Task 03

Write a program to: (a) prompt the user, (b) read the first initial, then the middle, and then the last initial of a person's name, and then (c) display them down the left margin.

Sample execution:

Enter First Initial: S
Enter Second Initial: F
Enter Third Initial: L
Enter Last initial: M

S
FL
M

edit: F:\9TH\341 LAB\LabA2_Task03.asm

edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
;Task 03
;Write a program to: (a) prompt the user, (b) read first initial, then the secon
;and then the third and finally last initial, and then (c) display them the way
.MODEL SMALL
.STACK 100H
.DATA
    ; DEFINE YOUR VARIABLES HERE
    P1 DB "Enter First Initial: $"
    P2 DB "Enter Second Initial: $"
    P3 DB "Enter Third Initial: $"
    P4 DB "Enter Last Initial: $"
.CODE
    MAIN PROC
        MOV AX, @DATA
        MOV DS, AX
        ; YOUR CODE STARTS HERE
        ;-----------------------------------------
        ;First we take 4 inputs using 4 prompts
        ;-----------------------------------------

        LEA DX,P1   ;load effective address
        MOV AH,9    ;string output
        INT 21H     ;interrupt
        MOV AH,1    ;string char input for First Initial
        INT 21H     ;interrupt
        MOV BH,AL   ;BH=AL=First Initial

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        LEA DX,P2   ;load effective address
        MOV AH,9    ;string output
        INT 21H     ;interrupt
        MOV AH,1    ;string char input for Second Initial
        INT 21H     ;interrupt
        MOV CH,AL   ;CH=AL=Second Initial

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        LEA DX,P3   ;load effective address
        MOV AH,9    ;string output
        INT 21H     ;interrupt
        MOV AH,1    ;string char input for Third Initial
        INT 21H     ;interrupt
        MOV CL,AL   ;CL=AL=Third Initial

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        LEA DX,P4   ;load effective address
        MOV AH,9    ;string output
        INT 21H     ;interrupt
        MOV AH,1    ;string char input for Last Initial
        INT 21H     ;interrupt
        MOV BL,AL   ;BL=AL=LastInitial

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt
```

```
073         ;-----------------------------------------
074         ;Now we print the outputs
075         ;-----------------------------------------
076
077
078         MOV DL,BH   ;DL=first initial
079         MOV AH,2    ;string char output
080         INT 21H     ;interrupt
081
082         MOV AH, 2
083         MOV DL, 0DH;carriage return
084         INT 21H     ;interrupt
085         MOV DL, 0AH;line feed
086         INT 21H     ;interrupt
087
088
089
090
091         MOV DL,CH   ;DL=second initial
092         MOV AH,2    ;string char output
093         INT 21H     ;interrupt
094         MOV DL,CL   ;DL=third initial
095         MOV AH,2    ;string char output
096         INT 21H     ;interrupt
097
098         MOV AH, 2
099         MOV DL, 0DH;carriage return
100         INT 21H     ;interrupt
101         MOV DL, 0AH;line feed
102         INT 21H     ;interrupt
103
104
105
106
107         MOV DL,BL   ;DL=Last initial
108         MOV AH,2    ;string char output
109         INT 21H     ;interrupt
110
111         ; YOUR CODE ENDS HERE
112
113         MOV AX, 4C00H
114         INT 21H
115
116     MAIN ENDP
117     END MAIN
```

;Task 03
;Write a program to: (a) prompt the user, (b) read first initial, then the second,
;and then the third and finally last initial, and then (c) display them the way Shown below.

```
.MODEL SMALL
.STACK 100H
.DATA
   ; DEFINE YOUR VARIABLES HERE
   P1 DB "Enter First Initial: $"
   P2 DB "Enter Second Initial: $"
   P3 DB "Enter Third Initial: $"
   P4 DB "Enter Last Initial: $"
.CODE
   MAIN PROC
      MOV AX, @DATA
      MOV DS, AX
      ; YOUR CODE STARTS HERE
```

```
;-------------------------------------------
;First we take 4 inputs using 4 prompts
;-------------------------------------------

LEA DX,P1  ;load effective address
MOV AH,9   ;string output
INT 21H    ;interrupt
MOV AH,1   ;string char input for First Initial
INT 21H    ;interrupt
MOV BH,AL  ;BH=AL=First Initial

MOV AH, 2
MOV DL, 0DH;carriage return
INT 21H    ;interrupt
MOV DL, 0AH;line feed
INT 21H    ;interrupt

LEA DX,P2  ;load effective address
MOV AH,9   ;string output
INT 21H    ;interrupt
MOV AH,1   ;string char input for Second Initial
INT 21H    ;interrupt
MOV CH,AL  ;CH=AL=Second Initial

MOV AH, 2
MOV DL, 0DH;carriage return
INT 21H    ;interrupt
MOV DL, 0AH;line feed
INT 21H    ;interrupt

LEA DX,P3  ;load effective address
MOV AH,9   ;string output
INT 21H    ;interrupt
MOV AH,1   ;string char input for Third Initial
INT 21H    ;interrupt
MOV CL,AL  ;CL=AL=Third Initial

MOV AH, 2
MOV DL, 0DH;carriage return
INT 21H    ;interrupt
MOV DL, 0AH;line feed
INT 21H    ;interrupt

LEA DX,P4  ;load effective address
MOV AH,9   ;string output
INT 21H    ;interrupt
MOV AH,1   ;string char input for Last Initial
INT 21H    ;interrupt
MOV BL,AL  ;BL=AL=LastInitial
```

```
    MOV AH, 2
    MOV DL, 0DH;carriage return
    INT 21H    ;interrupt
    MOV DL, 0AH;line feed
    INT 21H    ;interrupt

    ;-------------------------------------------
    ;Now we print the outputs
    ;-------------------------------------------


    MOV DL,BH  ;DL=first initial
    MOV AH,2   ;string char output
    INT 21H    ;interrupt

    MOV AH, 2
    MOV DL, 0DH;carriage return
    INT 21H    ;interrupt
    MOV DL, 0AH;line feed
    INT 21H    ;interrupt

    MOV DL,CH  ;DL=second initial
    MOV AH,2   ;string char output
    INT 21H    ;interrupt
    MOV DL,CL  ;DL=third initial
    MOV AH,2   ;string char output
    INT 21H    ;interrupt

    MOV AH, 2
    MOV DL, 0DH;carriage return
    INT 21H    ;interrupt
    MOV DL, 0AH;line feed
    INT 21H    ;interrupt

    MOV DL,BL  ;DL=Last initial
    MOV AH,2   ;string char output
    INT 21H    ;interrupt

    ; YOUR CODE ENDS HERE
    MOV AX, 4C00H
    INT 21H

MAIN ENDP
END MAIN
```

Task 04

Write a program to read two hex digits A-F, and then display the subtraction on the next line in decimal.

Sample execution:

ENTER FIRST HEX DIGIT: C
ENTER SECOND HEX DIGIT: A
IN DECIMAL SUBTRACTION IS 2

---

edit: F:\9TH\341 LAB\LabA2_Task04.asm

le edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator convertor options help about

```
;Task 04
;Write a program to read two hex digits A-F, and them and
;display the subtraction on the next line in decimal.

.MODEL SMALL
.STACK 100H
.DATA
    ; DEFINE YOUR VARIABLES HERE
    P1 DB "ENTER FIRST HEX DIGIT: $"
    P2 DB "ENTER SECOND HEX DIGIT: $"
    R DB "IN DECIMAL SUBTRACTION IS $"

.CODE
    MAIN PROC

        MOV AX, @DATA
        MOV DS, AX
        ; YOUR CODE STARTS HERE

        ;-----------------------------------------------
        ;First we take 2 Hex Digits using 2 prompts
        ;-----------------------------------------------

        LEA DX,P1   ;load effective address
        MOV AH,9    ;string output
        INT 21H     ;interrupt
        MOV AH,1    ;string char input for First HEX DIGIT
        INT 21H     ;interrupt
        MOV BH,AL   ;BH=AL=First HEX DIGIT
        SUB BH,11H

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        LEA DX,P2   ;load effective address
        MOV AH,9    ;string output
        INT 21H     ;interrupt
        MOV AH,1    ;string char input for Second HEX DIGIT
        INT 21H     ;interrupt
        MOV BL,AL   ;BL=AL=Second HEX DIGIT
        SUB BL,11H

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H     ;interrupt
        MOV DL, 0AH;line feed
        INT 21H     ;interrupt

        ;-----------------------------------------------
        ;Now we print the subtraction on the next line in decimal.
        ;-----------------------------------------------


        LEA DX,R   ;load effective address
        MOV AH,9    ;string output
        INT 21H

        SUB BH,BL   ;BH=BH-BL=1st hex-2nd hex (assuming 1st hex)>=2nd hex)
        MOV DL,BH   ;DL=BH= 1st hex-2nd hex
        ADD DL,30H  ;DL=DL+30H


        MOV AH,2
        INT 21H     ;interrupt

        ; YOUR CODE ENDS HERE

        MOV AX, 4C00H
        INT 21H

    MAIN ENDP
    END MAIN
```

```
;Task 04
;Write a program to read two hex digits A-F, and them and
;display the subtraction on the next line in decimal.

.MODEL SMALL
.STACK 100H
.DATA
    ; DEFINE YOUR VARIABLES HERE
    P1 DB "ENTER FIRST HEX DIGIT: $"
    P2 DB "ENTER SECOND HEX DIGIT: $"
    R DB "IN DECIMAL SUBTRACTION IS $"

.CODE
    MAIN PROC

        MOV AX, @DATA
        MOV DS, AX
        ; YOUR CODE STARTS HERE

        ;-----------------------------------------
        ;First we take 2 Hex Digits using 2 prompts
        ;-----------------------------------------

        LEA DX,P1  ;load effective address
        MOV AH,9   ;string output
        INT 21H    ;interrupt
        MOV AH,1   ;string char input for First HEX DIGIT
        INT 21H    ;interrupt
        MOV BH,AL  ;BH=AL=First HEX DIGIT
        SUB BH,11H

        MOV AH, 2
        MOV DL, 0DH;carriage return
        INT 21H    ;interrupt
        MOV DL, 0AH;line feed
        INT 21H    ;interrupt

        LEA DX,P2  ;load effective address
        MOV AH,9   ;string output
        INT 21H    ;interrupt
        MOV AH,1   ;string char input for Second HEX DIGIT
        INT 21H    ;interrupt
        MOV BL,AL  ;BL=AL=Second HEX DIGIT
        SUB BL,11H
```

```
    MOV AH, 2
    MOV DL, 0DH;carriage return
    INT 21H    ;interrupt
    MOV DL, 0AH;line feed
    INT 21H    ;interrupt

    ;---------------------------------------------
    ;Now we print the subtraction on the next line in decimal.
    ;---------------------------------------------

    LEA DX,R  ;load effective address
    MOV AH,9   ;string output
    INT 21H

    SUB BH,BL  ;BH=BH-BL=1st hex-2nd hex (assuming 1st hex>=2nd hex)
    MOV DL,BH  ;DL=BH= 1st hex-2nd hex
    ADD DL,30H ;DL=DL+30H

    MOV AH,2
    INT 21H    ;interrupt

    ; YOUR CODE ENDS HERE

    MOV AX, 4C00H
    INT 21H

MAIN ENDP
END MAIN
```