

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

edit bookmarks assembler emulator math ascii codes help

```

new open examples save compile emulate calculator convertor options help about
01 ;Task 01
02 ;Take two numbers as input, add them, print the message "The result is" and then
03 ;display the result in the next line.
04
05 .MODEL SMALL
06
07 .STACK 100H
08
09 .DATA
10 ; DEFINE YOUR VARIABLES HERE
11 S DB "The result is $"
12 .CODE
13 MAIN PROC
14
15     MOV AX, @DATA
16     MOV DS, AX
17
18     ; YOUR CODE STARTS HERE
19
20
21     MOV AH, 1 ;string char input for num 1
22     INT 21H ;interrupt
23     MOV BL, AL ;BL=AL=num 1 input
24
25     MOV AH, 2
26     MOV DL, 0DH ;carriage return
27     INT 21H ;interrupt
28     MOV DL, 0AH ;line feed
29     INT 21H ;interrupt
30
31     MOV AH, 1 ;string char input for num 2
32     INT 21H ;interrupt
33
34     ADD BL, AL ;BL = BL+AL = num1+ num2
35
36     MOV AH, 2
37     MOV DL, 0DH ;carriage return
38     INT 21H ;interrupt
39     MOV DL, 0AH ;line feed
40     INT 21H ;interrupt
41
42     LEA DX, S ;load effective address
43     MOV AH, 9 ;string output
44     INT 21H ;interrupt
45
46     MOV AH, 2
47     MOV DL, 0DH ;carriage return
48     INT 21H ;interrupt
49     MOV DL, 0AH ;line feed
50     INT 21H ;interrupt
51
52     MOV DL, BL ;DL = BL = num1 + num2
53     SUB DL, 30H ;DL=DL-30H= num1+num2-30H
54
55     MOV AH, 2 ;string char output
56     INT 21H ;interrupt
57
58     ; YOUR CODE ENDS HERE
59
60     MOV AX, 4C00H
61     INT 21H
62
63 MAIN ENDP
64 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

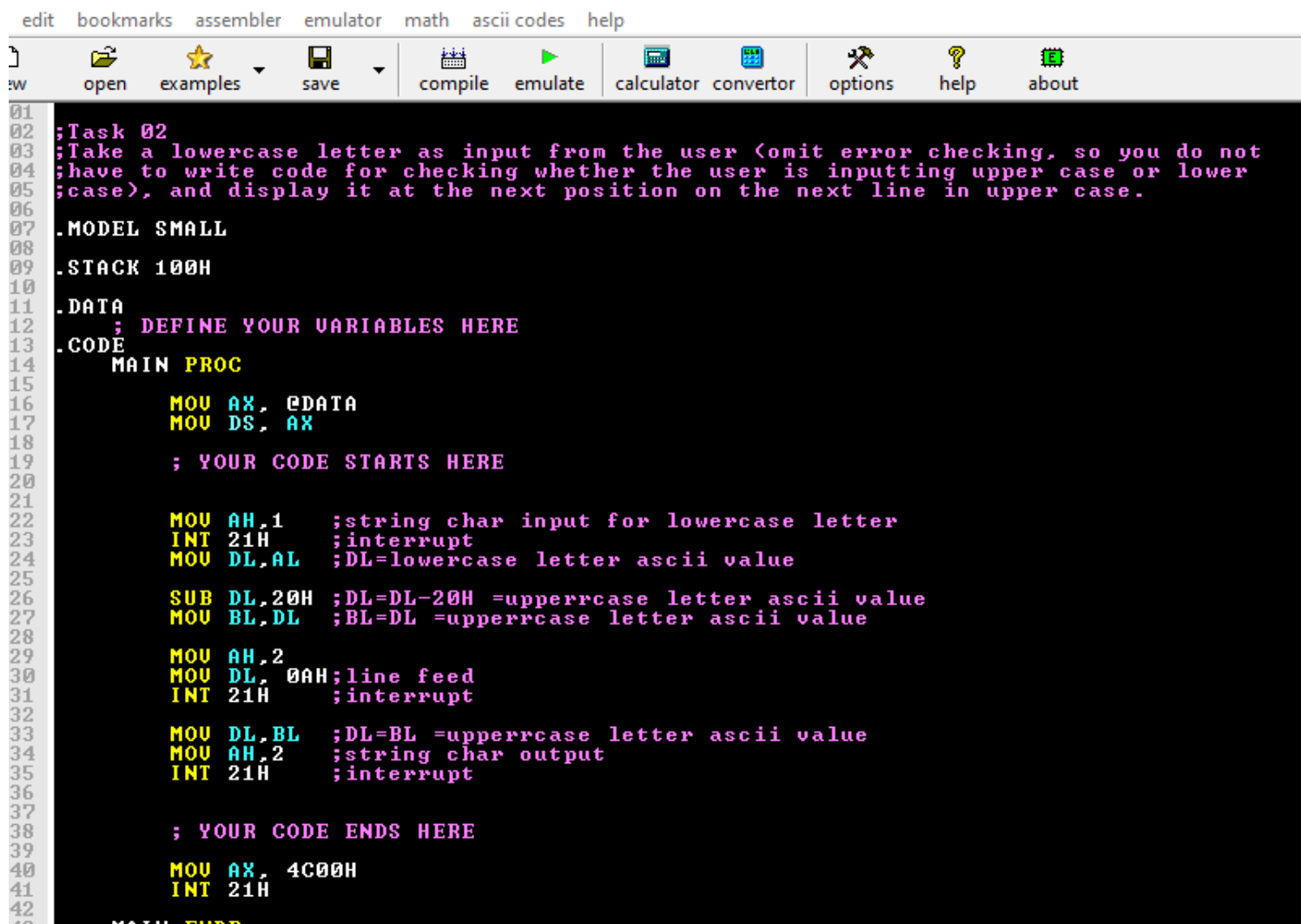
A

Sample Execution 2

b

B

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



```
01 ;Task 02
02 ;Take a lowercase letter as input from the user (omit error checking, so you do not
03 ;have to write code for checking whether the user is inputting upper case or lower
04 ;case), and display it at the next position on the next line in upper case.
05
06 .MODEL SMALL
07
08 .STACK 100H
09
10 .DATA
11 ; DEFINE YOUR VARIABLES HERE
12
13 .CODE
14 MAIN PROC
15
16     MOV AX, 0DATA
17     MOV DS, AX
18
19     ; YOUR CODE STARTS HERE
20
21
22     MOV AH,1 ;string char input for lowercase letter
23     INT 21H ;interrupt
24     MOV DL,AL ;DL=lowercase letter ascii value
25
26     SUB DL,20H ;DL=DL-20H =upperrcase letter ascii value
27     MOV BL,DL ;BL=DL =upperrcase letter ascii value
28
29     MOV AH,2
30     MOV DL, 0AH;line feed
31     INT 21H ;interrupt
32
33     MOV DL,BL ;DL=BL =upperrcase letter ascii value
34     MOV AH,2 ;string char output
35     INT 21H ;interrupt
36
37     ; YOUR CODE ENDS HERE
38
39
40     MOV AX, 4C00H
41     INT 21H
42 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

```

0001 ;Task 03
0002 ;Write a program to: (a) prompt the user, (b) read first initial, then the second
0003 ;and then the third and finally last initial, and then (c) display them the way
0004 .MODEL SMALL
0005 .STACK 100H
0006 .DATA
0007 ; DEFINE YOUR VARIABLES HERE
0008 P1 DB "Enter First Initial: $"
0009 P2 DB "Enter Second Initial: $"
0010 P3 DB "Enter Third Initial: $"
0011 P4 DB "Enter Last Initial: $"
0012 .CODE
0013 MAIN PROC
0014     MOV AX, @DATA
0015     MOV DS, AX
0016     ; YOUR CODE STARTS HERE
0017     ;-----
0018     ;First we take 4 inputs using 4 prompts
0019     ;-----
0020
0021     LEA DX,P1 ;load effective address
0022     MOV AH,9 ;string output
0023     INT 21H ;interrupt
0024     MOV AH,1 ;string char input for First Initial
0025     INT 21H ;interrupt
0026     MOV BH,AL ;BH=AL=First Initial
0027
0028     MOV AH, 2
0029     MOV DL, 0DH;carriage return
0030     INT 21H ;interrupt
0031     MOV DL, 0AH;line feed
0032     INT 21H ;interrupt
0033
0034     LEA DX,P2 ;load effective address
0035     MOV AH,9 ;string output
0036     INT 21H ;interrupt
0037     MOV AH,1 ;string char input for Second Initial
0038     INT 21H ;interrupt
0039     MOV CH,AL ;CH=AL=Second Initial
0040
0041     MOV AH, 2
0042     MOV DL, 0DH;carriage return
0043     INT 21H ;interrupt
0044     MOV DL, 0AH;line feed
0045     INT 21H ;interrupt
0046
0047     LEA DX,P3 ;load effective address
0048     MOV AH,9 ;string output
0049     INT 21H ;interrupt
0050     MOV AH,1 ;string char input for Third Initial
0051     INT 21H ;interrupt
0052     MOV CL,AL ;CL=AL=Third Initial
0053
0054     MOV AH, 2
0055     MOV DL, 0DH;carriage return
0056     INT 21H ;interrupt
0057     MOV DL, 0AH;line feed
0058     INT 21H ;interrupt
0059
0060     LEA DX,P4 ;load effective address
0061     MOV AH,9 ;string output
0062     INT 21H ;interrupt
0063     MOV AH,1 ;string char input for Last Initial
0064     INT 21H ;interrupt
0065     MOV BL,AL ;BL=AL=LastInitial
0066
0067     MOV AH, 2
0068     MOV DL, 0DH;carriage return
0069     INT 21H ;interrupt
0070     MOV DL, 0AH;line feed
0071     INT 21H ;interrupt
0072

```

```

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

01 ;Task 04
02 ;Write a program to read two hex digits A-F, and then and
03 ;display the subtraction on the next line in decimal.
04
05 .MODEL SMALL
06 .STACK 100H
07 .DATA
08 ; DEFINE YOUR VARIABLES HERE
09 P1 DB "ENTER FIRST HEX DIGIT: $"
10 P2 DB "ENTER SECOND HEX DIGIT: $"
11 R DB "IN DECIMAL SUBTRACTION IS $"
12
13 .CODE
14 MAIN PROC
15
16     MOV AX, @DATA
17     MOV DS, AX
18     ; YOUR CODE STARTS HERE
19
20     ;-----
21     ;First we take 2 Hex Digits using 2 prompts
22     ;-----
23
24     LEA DX,P1 ;load effective address
25     MOV AH,9 ;string output
26     INT 21H ;interrupt
27     MOV AH,1 ;string char input for First HEX DIGIT
28     INT 21H ;interrupt
29     MOV BH,AL ;BH=AL=First HEX DIGIT
30     SUB BH,11H
31
32     MOV AH, 2
33     MOV DL, 0DH;carriage return
34     INT 21H ;interrupt
35     MOV DL, 0AH;line feed
36     INT 21H ;interrupt
37
38     LEA DX,P2 ;load effective address
39     MOV AH,9 ;string output
40     INT 21H ;interrupt
41     MOV AH,1 ;string char input for Second HEX DIGIT
42     INT 21H ;interrupt
43     MOV BL,AL ;BL=AL=Second HEX DIGIT
44     SUB BL,11H
45
46     MOV AH, 2
47     MOV DL, 0DH;carriage return
48     INT 21H ;interrupt
49     MOV DL, 0AH;line feed
50     INT 21H ;interrupt
51
52     ;-----
53     ;Now we print the subtraction on the next line in decimal.
54     ;-----
55
56
57     LEA DX,R ;load effective address
58     MOV AH,9 ;string output
59     INT 21H
60
61     SUB BH,BL ;BH=BH-BL=1st hex-2nd hex (assuming 1st hex>=2nd hex)
62     MOV DL,BH ;DL=BH= 1st hex-2nd hex
63     ADD DL,30H ;DL=DL+30H
64
65
66     MOV AH,2
67     INT 21H ;interrupt
68
69     ; YOUR CODE ENDS HERE
70
71     MOV AX, 4C00H
72     INT 21H
73
74 MAIN ENDP
75 END MAIN
76
```

;Task 04

;Write a program to read two hex digits A-F, and then 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
```