# Riphah School of Computing and Innovation (RSCI), Lahore



Computer Organization and Assembly Language (Lab)

## Lab Report # 3

**Submitted to:** Ms. Amna Bibi

**Submitted by:**

{Muhammad Abdullah}

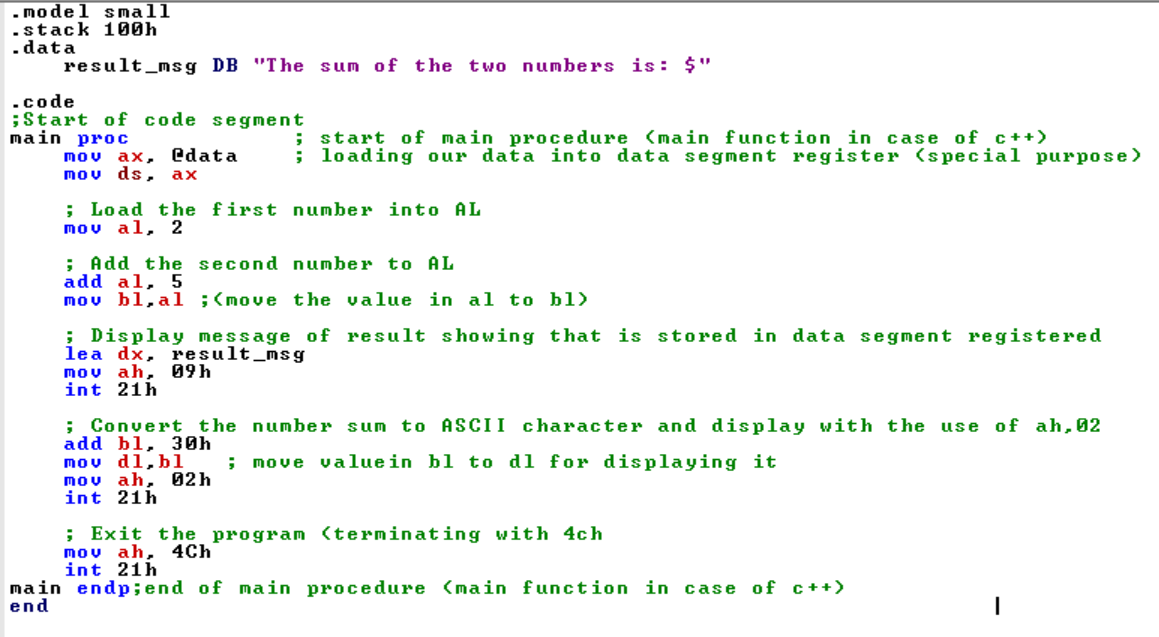
{53457}

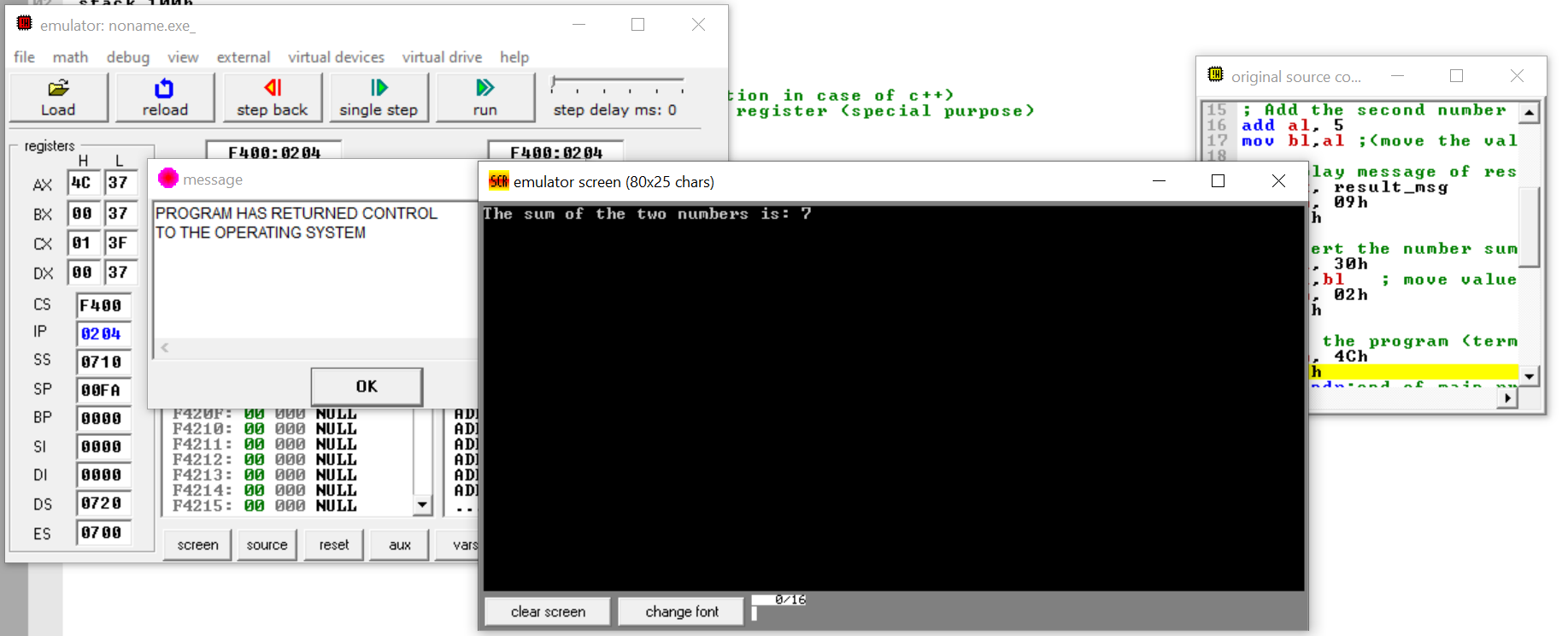
{BSCS-3B}

{16/11/2024}

### Addition Hard Code

|  |
| --- |
| .model small  .stack 100h  .data  result\_msg DB "The sum of the two numbers is: $"  .code  ;Start of code segment  main proc ; start of main procedure (main function in case of c++)  mov ax, @data ; loading our data into data segment register (special purpose)  mov ds, ax  ; Load the first number into AL  mov al, 2  ; Add the second number to AL  add al, 5  mov bl,al ;(move the value in al to bl)  ; Display message of result showing that is stored in data segment registered  lea dx, result\_msg  mov ah, 09h  int 21h  ; Convert the number sum to ASCII character and display with the use of ah,02  add bl, 30h  mov dl,bl ; move valuein bl to dl for displaying it  mov ah, 02h  int 21h  ; Exit the program (terminating with 4ch  mov ah, 4Ch  int 21h  main endp;end of main procedure (main function in case of c++)  end |



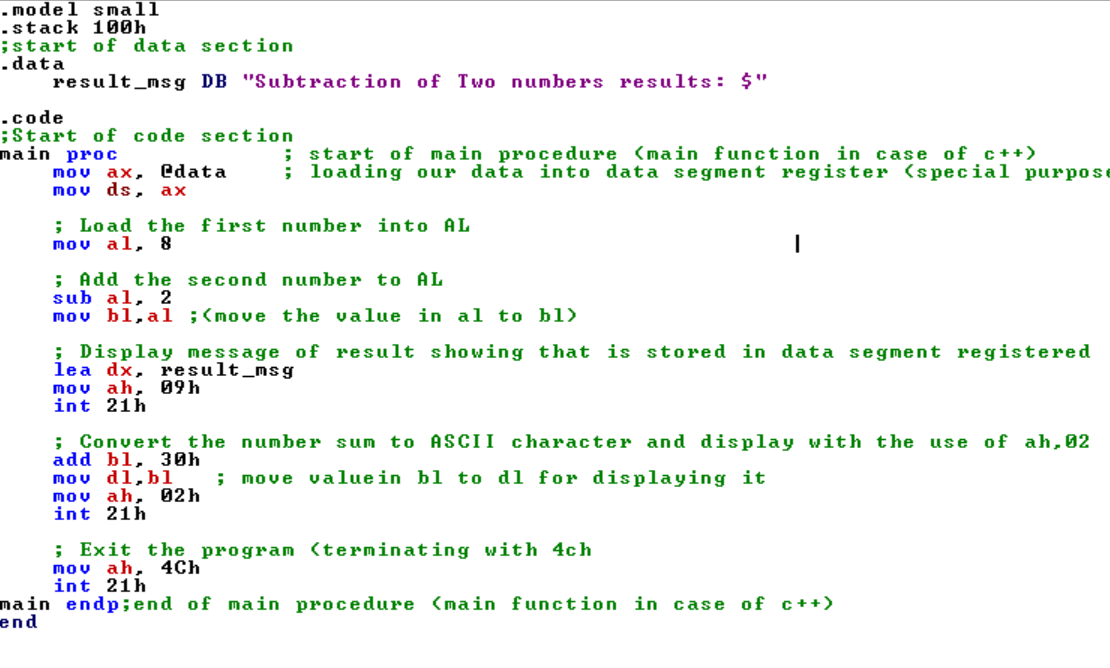


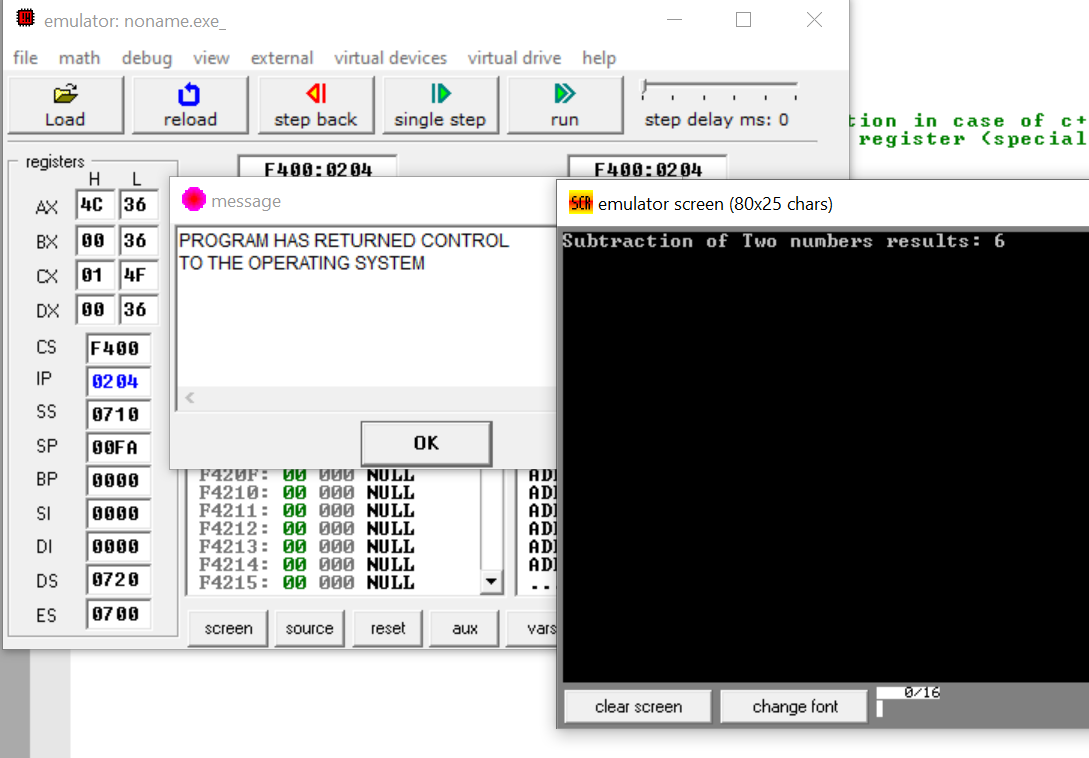
**Explanation:**

* Defining the variable result message the data section.
* Main procedure (main function in case of c++)
* Load all data section variables into data segment register.
* Loading 2 to AL.
* Adding 5 to AL (2) now AL after addition is 7
* The result 7 is then loaded into BL.
* Added 30h to BL for converting number back into the character.
* The ASCII value of 7 is then loaded into DL for displaying.
* AH,02h is used to display the character stored in DL (which is '7').
* Exit Program with 4ch
* End main procedure
* End program

### Difference Hard Code

|  |
| --- |
| .model small  .stack 100h  ;start of data section  .data  result\_msg DB "Subtraction of Two numbers results: $"  .code  ;Start of code section  main proc ; start of main procedure (main function in case of c++)  mov ax, @data ; loading our data into data segment register (special purpose)  mov ds, ax  ; Load the first number into AL  mov al, 8  ; Add the second number to AL  sub al, 2  mov bl,al ;(move the value in al to bl)  ; Display message of result showing that is stored in data segment registered  lea dx, result\_msg  mov ah, 09h  int 21h  ; Convert the number sum to ASCII character and display with the use of ah,02  add bl, 30h  mov dl,bl ; move valuein bl to dl for displaying it  mov ah, 02h  int 21h  ; Exit the program (terminating with 4ch  mov ah, 4Ch  int 21h  main endp;end of main procedure (main function in case of c++)  end |



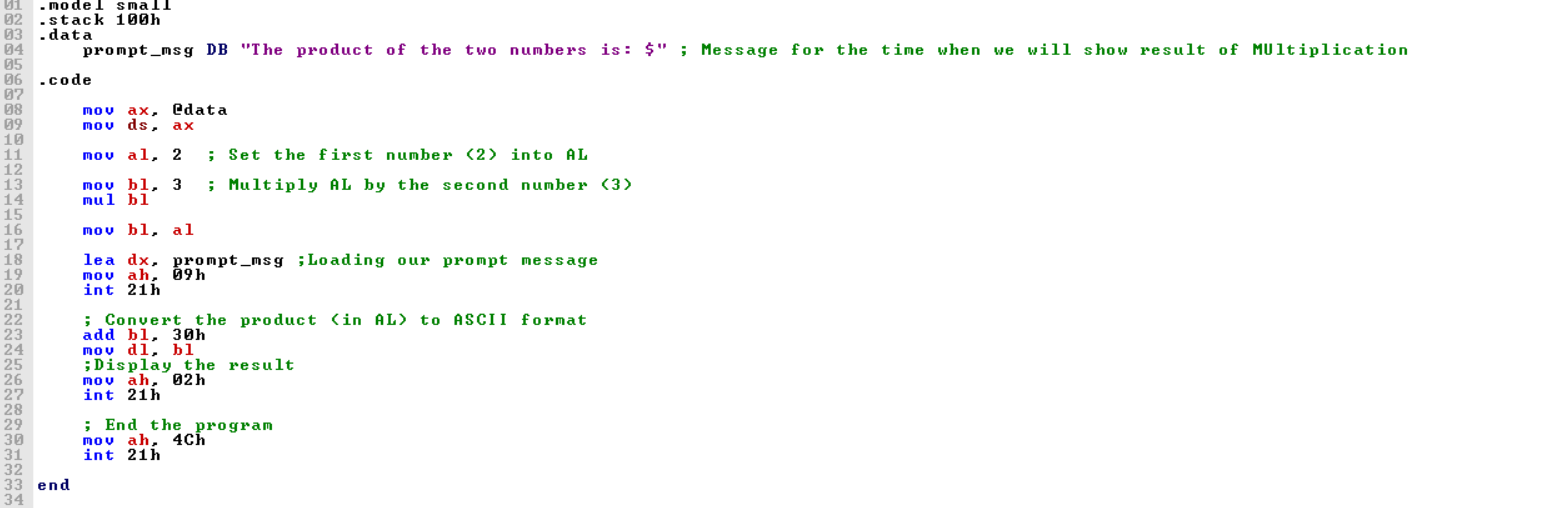


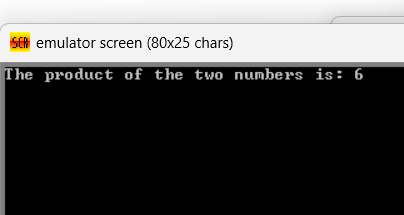
**Explanation:**

* Defining the variable result message in the data section.
* Main procedure (main function in case of c++)
* Load all data section variables into data segment register.
* Loading 2 to AL.
* Adding 5 to AL (2) now AL after addition is 7
* The result 7 is then loaded into BL.
* Added 30h to BL for converting number back into the character.
* The ASCII value of 7 is then loaded into DL for displaying.
* AH,02h is used to display the character stored in DL (which is '7').
* Exit Program with 4ch
* End main procedure
* End program

### Product

|  |
| --- |
| .model small  .stack 100h  .data  prompt\_msg DB "The product of the two numbers is: $" ; Message for the time when we will show result of MUltiplication  .code  mov ax, @data  mov ds, ax    mov al, 2 ; Set the first number (2) into AL  mov bl, 3 ; Multiply AL by the second number (3)  mul bl    mov bl, al  lea dx, prompt\_msg ;Loading our prompt message  mov ah, 09h  int 21h  ; Convert the product (in AL) to ASCII format  add bl, 30h  mov dl, bl  ;Display the result  mov ah, 02h  int 21h  ; End the program  mov ah, 4Ch  int 21h  end |



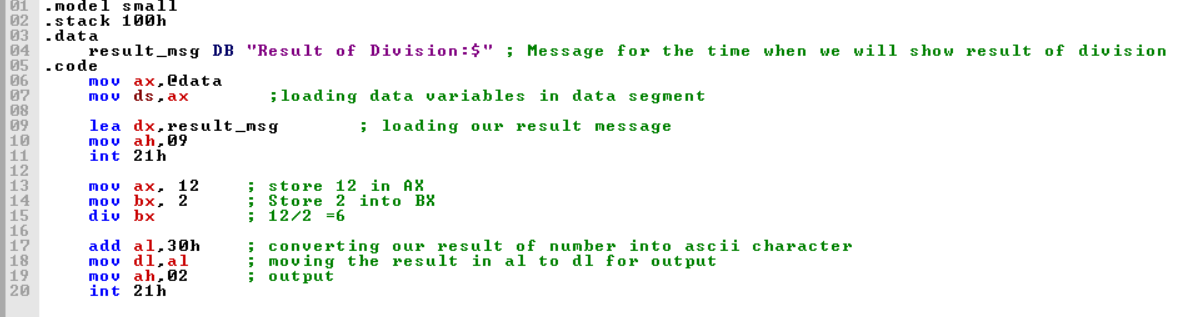


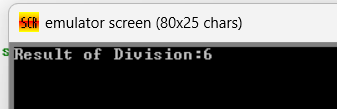
**Explanation:**

* Defining Result prompt message in Data Segment
* Output of Result prompt message using 09H command.
* Preparation for Multiplication by loading `AX` with `2` and `BX` set to `3`. (i.e. value in `AX` is to be multiplied with the value in `BX`).
* `MUL BX`; this instruction divides the contents of `AX` by `BX`. Here, the content in `AX` is `2` while the content in `BX` equals `3`, therefore the result (`6`) which is stored in `AL`.
* Mov BL,AL makes the value ‘6’ in AL stored in BL.
* `ADD BL, 30h` makes the output 6 a number, represent `6` in ASCII as a character
* Moving BL’s value to our DL for displaying purpose.
* The character coded for `6` is transferred into `DL` and displayed on screen using 02h command.

### Division

|  |
| --- |
| .model small  .stack 100h  .data  result\_msg DB "Result of Division:$" ; Message for the time when we will show result of division  .code  mov ax,@data  mov ds,ax ;loading data variables in data segment    lea dx,result\_msg ; loading our result message  mov ah,09  int 21h    mov ax, 12 ; store 12 in AX  mov bx, 2 ; Store 2 into BX  div bx ; 12/2 =6    add al,30h ; converting our result of number into ascii character  mov dl,al ; moving the result in al to dl for output  mov ah,02 ; output  int 21h |





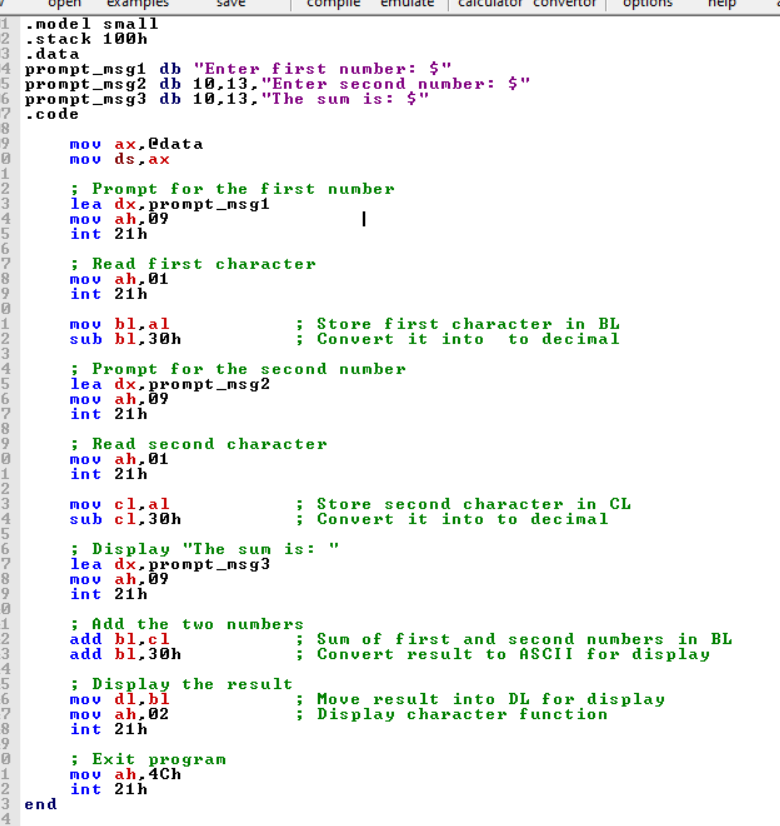
**Explanation:**

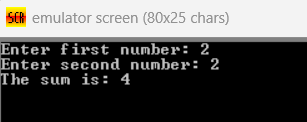
* Defining Result prompt message in Data Segment
* Output of Result prompt message using 09H command
* Preparation for division by loading `AX` with `12` and `BX` set to `2`. (i.e. value in `AX` is to be divided by the value in `BX`).
* `DIV BX`; this instruction divides the contents of `AX` by `BX`. Here, the content in `AX` is `12` while the content in `BX` equals `2`, therefore the result (`6`) is stored in `AL`.
* `ADD AL, 30h` makes the output 6 a number, represent `6` in ASCII as a character.
* The character coded for `6` is transferred into `DL` and displayed on screen using 02h command.

# User Input:

### Sum

|  |
| --- |
| .model small  .stack 100h  .data  prompt\_msg1 db "Enter first number: $"  prompt\_msg2 db 10,13,"Enter second number: $"  prompt\_msg3 db 10,13,"The sum is: $"  .code  mov ax,@data  mov ds,ax    ; Prompt for the first number  lea dx,prompt\_msg1  mov ah,09  int 21h    ; Read first character  mov ah,01  int 21h    mov bl,al ; Store first character in BL  sub bl,30h ; Convert it into to decimal    ; Prompt for the second number  lea dx,prompt\_msg2  mov ah,09  int 21h    ; Read second character  mov ah,01  int 21h    mov cl,al ; Store second character in CL  sub cl,30h ; Convert it into to decimal    ; Display "The sum is: "  lea dx,prompt\_msg3  mov ah,09  int 21h    ; Add the two numbers  add bl,cl ; Sum of first and second numbers in BL  add bl,30h ; Convert result to ASCII for display    ; Display the result  mov dl,bl ; Move result into DL for display  mov ah,02 ; Display character function  int 21h    ; Exit program  mov ah,4Ch  int 21h  end |



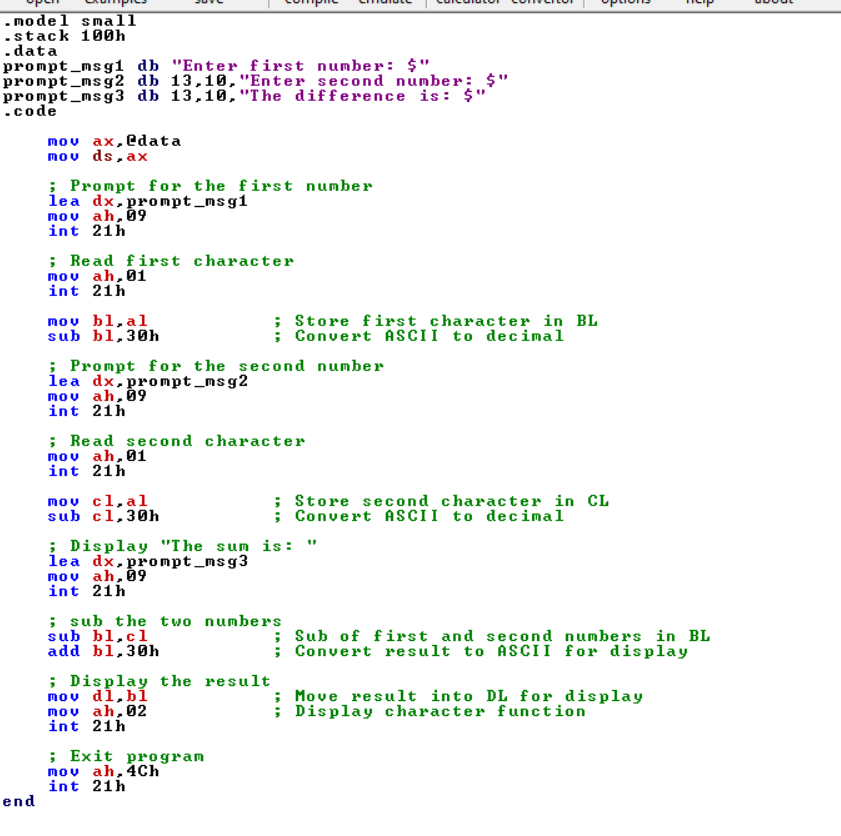


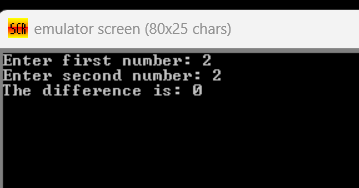
**Explanation:**

* First we define three prompt messages: one for the first number, one for the second number, and one for displaying the sum.
* We have used 09 and lea for displaying these prompt messages on screen.
* After first prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to BL and converts it from ASCII to decimal by using sub bl, 30h.
* After Second prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to CL and converts it from ASCII to decimal by using sub cl, 30h.
* After this we display the "The sum is: " message, adds the two numbers in bl and cl
* We then convert the result to ASCII using add bl,30h, and move our value of asci that is converted freshly from bl to dl for displaying it on screen using 02 command.
* Finally, it exits the program using 4Ch.

### Difference

|  |
| --- |
| .model small  .stack 100h  .data  prompt\_msg1 db "Enter first number: $"  prompt\_msg2 db 13,10,"Enter second number: $"  prompt\_msg3 db 13,10,"The difference is: $"  .code  mov ax,@data  mov ds,ax    ; Prompt for the first number  lea dx,prompt\_msg1  mov ah,09  int 21h    ; Read first character  mov ah,01  int 21h    mov bl,al ; Store first character in BL  sub bl,30h ; Convert ASCII to decimal    ; Prompt for the second number  lea dx,prompt\_msg2  mov ah,09  int 21h    ; Read second character  mov ah,01  int 21h    mov cl,al ; Store second character in CL  sub cl,30h ; Convert ASCII to decimal    ; Display "The sum is: "  lea dx,prompt\_msg3  mov ah,09  int 21h    ; sub the two numbers  sub bl,cl ; Sub of first and second numbers in BL  add bl,30h ; Convert result to ASCII for display    ; Display the result  mov dl,bl ; Move result into DL for display  mov ah,02 ; Display character function  int 21h    ; Exit program  mov ah,4Ch  int 21h  end |



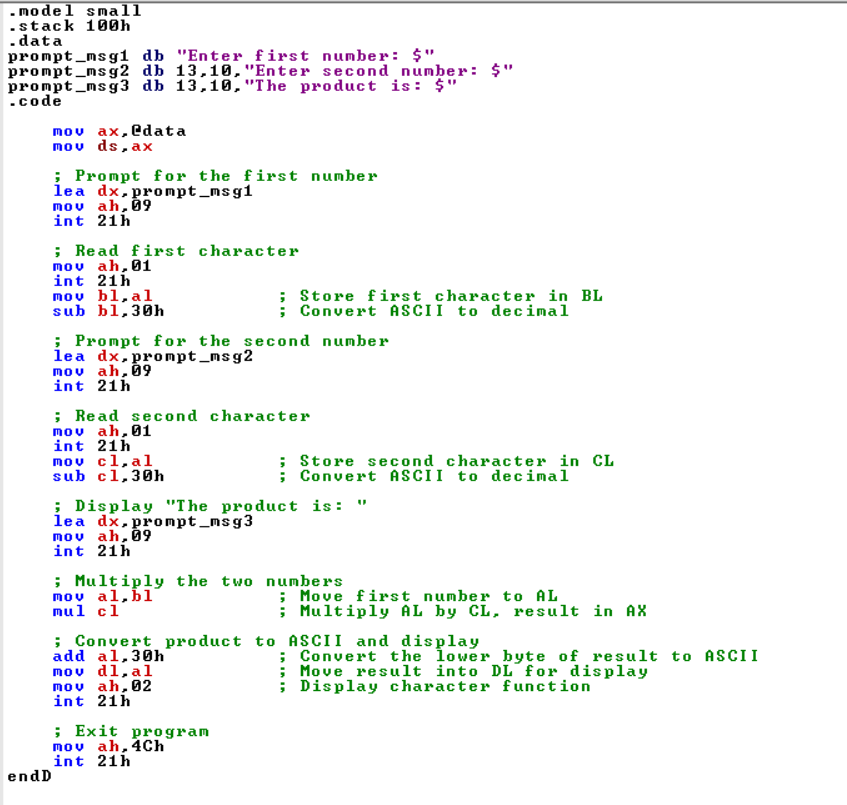


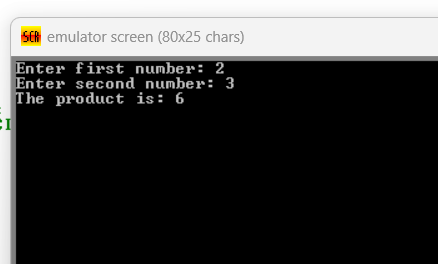
**Explanation:**

* First we define three prompt messages: one for the first number, one for the second number, and one for displaying the Difference.
* We have used 09 and lea for displaying these prompt messages on screen.
* After first prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to BL and converts it from ASCII to decimal by using sub bl, 30h.
* After Second prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to CL and converts it from ASCII to decimal by using sub cl, 30h.
* After this we display the "The Difference is: " message, Subtract number in cl from bl
* We then convert the result to ASCII using add bl,30h, and move our value of asci that is converted freshly from bl to dl for displaying it on screen using 02 command.
* Finally, it exits the program using 4Ch.

### Product

|  |
| --- |
| .model small  .stack 100h  .data  prompt\_msg1 db "Enter first number: $"  prompt\_msg2 db 13,10,"Enter second number: $"  prompt\_msg3 db 13,10,"The product is: $"  .code  mov ax,@data  mov ds,ax  ; Prompt for the first number  lea dx,prompt\_msg1  mov ah,09  int 21h  ; Read first character  mov ah,01  int 21h  mov bl,al ; Store first character in BL  sub bl,30h ; Convert ASCII to decimal    ; Prompt for the second number  lea dx,prompt\_msg2  mov ah,09  int 21h  ; Read second character  mov ah,01  int 21h  mov cl,al ; Store second character in CL  sub cl,30h ; Convert ASCII to decimal    ; Display "The product is: "  lea dx,prompt\_msg3  mov ah,09  int 21h  ; Multiply the two numbers  mov al,bl ; Move first number to AL  mul cl ; Multiply AL by CL, result in AX  ; Convert product to ASCII and display  add al,30h ; Convert the lower byte of result to ASCII  mov dl,al ; Move result into DL for display  mov ah,02 ; Display character function  int 21h  ; Exit program  mov ah,4Ch  int 21h  end |



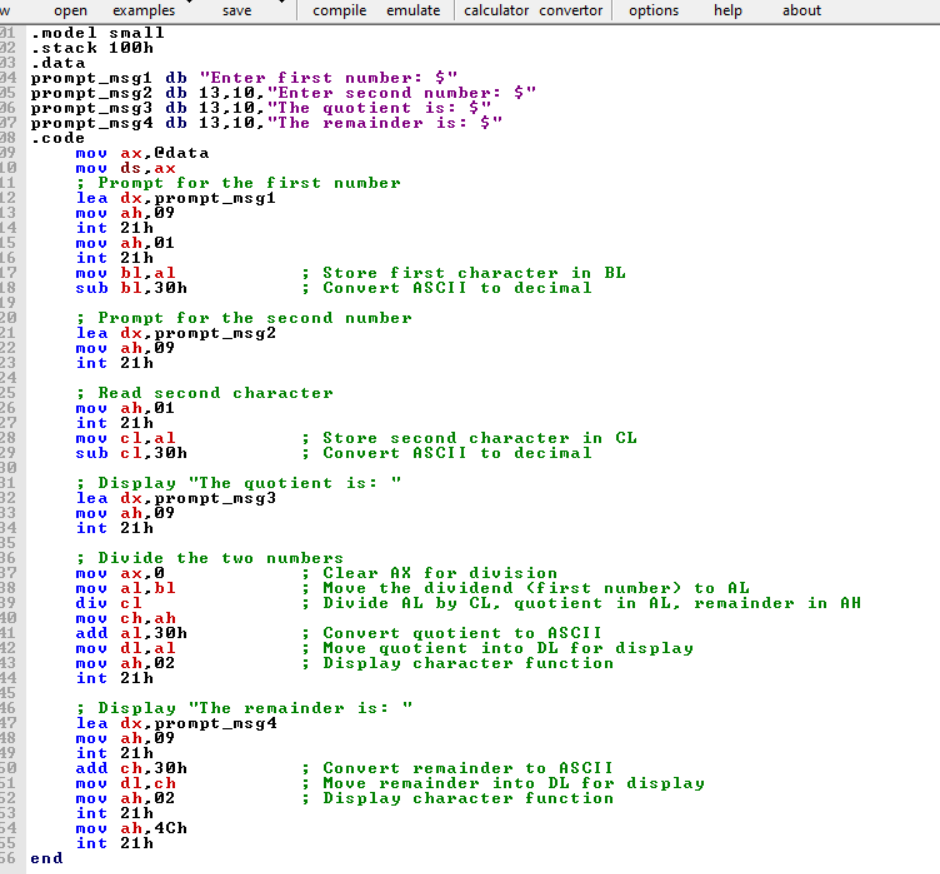


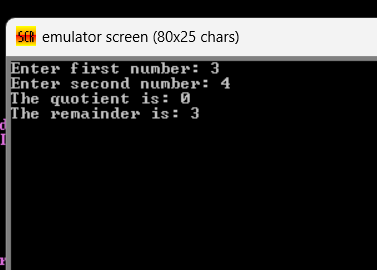
**Explanation:**

* First we define three prompt messages: one for the first number, one for the second number, and one for displaying the Product.
* We have used 09 and lea for displaying these prompt messages on screen.
* After first prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to BL and converts it from ASCII to decimal by using sub bl, 30h.
* After Second prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to CL and converts it from ASCII to decimal by using sub cl, 30h.
* After this we display the "The Product is: " message.
* We load our value in BL to AL using MOV AL,BL, then we use MUL CL to multiply value in CL with value in AL and store it in AL
* We then convert the result to ASCII using add al,30h, and move our value of asci that is converted freshly from al to dl for displaying it on screen using 02 command.
* Finally, it exits the program using 4Ch.

### Division

|  |
| --- |
| .model small  .stack 100h  .data  prompt\_msg1 db "Enter first number: $"  prompt\_msg2 db 13,10,"Enter second number: $"  prompt\_msg3 db 13,10,"The quotient is: $"  prompt\_msg4 db 13,10,"The remainder is: $"  .code  main proc  mov ax,@data  mov ds,ax  ; Prompt for the first number  lea dx,prompt\_msg1  mov ah,09  int 21h  ; Read first character  mov ah,01  int 21h  mov bl,al ; Store first character in BL  sub bl,30h ; Convert ASCII to decimal    ; Prompt for the second number  lea dx,prompt\_msg2  mov ah,09  int 21h  ; Read second character  mov ah,01  int 21h  mov cl,al ; Store second character in CL  sub cl,30h ; Convert ASCII to decimal  ; Display "The quotient is: "  lea dx,prompt\_msg3  mov ah,09  int 21h  ; Divide the two numbers  mov ax,0 ; Clear AX for division  mov al,bl ; Move the dividend (first number) to AL  div cl ; Divide AL by CL, quotient in AL, remainder in AH  mov ch,ah  ; Convert quotient to ASCII and display  add al,30h ; Convert quotient to ASCII  mov dl,al ; Move quotient into DL for display  mov ah,02 ; Display character function  int 21h  ; Display "The remainder is: "  lea dx,prompt\_msg4  mov ah,09  int 21h  ; Convert remainder to ASCII and display  add ch,30h ; Convert remainder to ASCII  mov dl,ch ; Move remainder into DL for display  mov ah,02 ; Display character function  int 21h  ; Exit program  mov ah,4Ch  int 21h  main endp  end main |

****



**Explanation:**

* First we define four prompt messages: one for the first number, one for the second number, aone for displaying the Quotient and last for displaying the remainder.
* We have used 09 and lea for displaying these prompt messages on screen.
* After first prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to BL and converts it from ASCII to decimal by using sub bl, 30h.
* After Second prompt message is displayed using 09h we take a single character input using 01h, stores it in AL, then moves the value to CL and converts it from ASCII to decimal by using sub cl, 30h.
* After this we display the "The Quotient is: " message.
* We first make our ax set to 0 to be on the safe side. Then we load our value(dividend) in BL to AL using MOV AL,BL, then we use DIV CL to multiply value in CL with value in AL and store it in AL. Our Remainder is Store in ah so we load our remainder from ah to ch in next line.
* We then convert the result to ASCII using add al,30h, and move our value of asci that is converted freshly from al to dl for displaying it on screen using 02 command.
* After this we display the "The Remainder is: " message
* We then convert the remainder in ch to ASCII using add ch,30h, and move our value of asci that is converted freshly from ch to dl for displaying it on screen using 02 command.
* Finally, it exits the program using 4Ch.