Assignment 3

Course Tittle: Microprocessor and Microcontroller Lab

Course Code: CSE3102

Section: 3

Submitted To

Shakib Mahmud Dipto

Lecturer, Dept of CSE

University of Liberal Arts Bangladesh

Submitted By:

MD. Jahidul Haque Junayer [221014071]

Spring, 24

Problem : Simple Program to add two numbers using EMU8086 Emulator

Code

```
include "emu8086.inc"
.model small
.stack 100h
.data
a db ?
.code
main proc
mov ax,@data ; for data
mov ds,ax
print "Enter a number: "
mov ah,1 ; takes input in al
int 21h ; interrupt
sub al, 48
mov a,al
mov dl,10 ; line feed
mov ah,2
int 21h
mov dl,13 ; carriage return
mov ah,2
int 21h
print "Enter another number: ";
mov ah,1
int 21h
sub al,48

add a,al
add a,48
mov dl,10 ; line feed
mov ah,2
int 21h
mov dl,13 ; carriage return
mov ah,2
int 21h

print "Addition : "
mov dl,a
mov ah,2 ; shows output from dl
int 21h ; interrupt
mov ah,4ch ; exit program
int 21h

main endp
end main
```

Code Explaination

Initialization:

include "emu8086.inc": This includes a file containing macros and other necessary definitions for working with the emu8086 assembler.

.model small: This directive sets the memory model to small, indicating that the code and data segments will fit within 64KB.

.stack 100h: Allocates 256 bytes (100h in hexadecimal) for the stack.

Data Segment:

.data section: Declares a variable a of type byte (db), which will be used to store the input numbers and their sum.

Code Segment:

.code section: Begins the code segment.

main procedure:

main proc: Declares the beginning of the main procedure.

Setting up Data Segment:

mov ax,@data: Loads the address of the data segment into the AX register.

mov ds,ax: Moves the value of AX into the DS register, setting up DS to point to the data segment.

Input of First Number:

print "Enter a number: ": Prints a prompt to the console.

mov ah,1: Sets up AH register to indicate function 1 of interrupt 21h (read character).

int 21h: Calls interrupt 21h to read a character from the standard input (keyboard). The input character is stored in AL register.

sub al, 48: Converts the ASCII character representing the digit to its corresponding numeric value and stores it in a.

Printing Line Feed and Carriage Return:

mov dl,10 and mov dl,13: Load the ASCII values for line feed (LF) and carriage return (CR) into DL register.

int 21h (twice): Calls interrupt 21h to print the LF and CR characters, producing a new line in the output.

Input of Second Number:

print "Enter another number: ": Prints a prompt to the console.

mov ah,1 and int 21h: Similar to the previous input process.

sub al,48: Converts the ASCII character representing the digit to its corresponding numeric value and subtracts 48 from AL to get the actual value.

Perform Addition:

add a,al: Adds the value of AL (the second number) to the variable a.

add a,48: Converts the sum back to ASCII representation.

mov dl,10 and mov dl,13: Load the ASCII values for LF and CR into DL register.

int 21h (twice): Prints LF and CR to produce a new line in the output.

Printing the Result:

print "Addition : ": Prints a prompt indicating the result.

mov dl,a: Moves the ASCII value of the sum stored in a to DL register.

mov ah,2 and int 21h: Calls interrupt 21h to print the value stored in DL.

Exit Program:

mov ah,4ch: Sets up AH register to indicate function 4Ch of interrupt 21h (exit program).

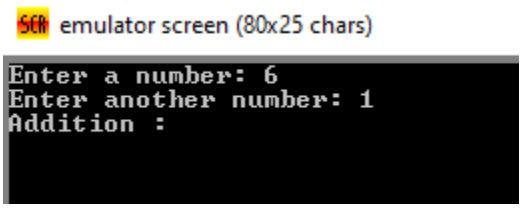
int 21h: Calls interrupt 21h to terminate the program.

End of main procedure and program:

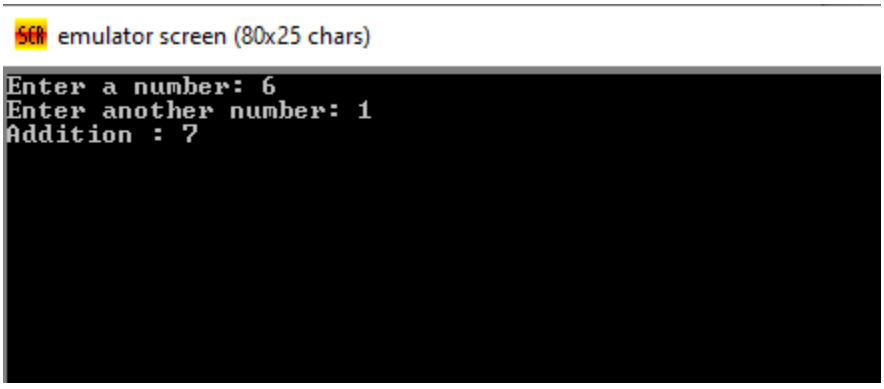
main endp: Marks the end of the main procedure.

end main: Marks the end of the program.

Input



Output



Problem : Simple Program to subtract two numbers using EMU8086 Emulator

Code

```
include "emu8086.inc"
.model small
.stack 100h

.data
    a db ?
    b db ?
.code
    newLine proc
        mov ah, 2
        mov bl, 10
        int 21h
        mov bl, 13
        int 21h
        ret
    main proc
        mov ax,@data
        mov ds,ax
        print "Enter the first number: "
        mov ah, 1
        int 21h
        mov a, al
        sub a, 48
        call newLine
        print "Enter the second number: "
        mov ah, 1
        int 21h
        mov b, al
        sub b, 48

        mov ch, a
        sub ch, b
        add ch, 48

        call newLine

        print "Subtraction: "

        mov ah, 2
        mov dl, ch
        int 21h

        Exit:
        mov ah, 4ch
        int 21h

        main endp
    end main
```

Code Explaination

Including header file:

include "emu8086.inc"

This line includes necessary libraries and macros for EMU8086.

Defining model and stack:

.model small

.stack 100h

Sets the memory model to "small" and allocates 100h (256 bytes) for the stack.

Data section:

.data

a db ?

b db ?

Declares two variables a and b of type byte.

Code section:

.code

Procedure newLine:

newLine proc

mov ah, 2

mov bl, 10

int 21h

mov bl, 13

int 21h

ret

newLine endp

This procedure prints a new line by calling DOS interrupt int 21h with AH=2 (print character). It first prints a line feed character (ASCII 10) and then a carriage return character (ASCII 13).

Procedure main:

main proc

This is the main procedure where the execution of the program starts.

Initializing Data Segment:

mov ax,@data

mov ds,ax

Initializes the data segment register ds with the address of the data segment.

Printing "Enter the first number: ":

print "Enter the first number: "

Prints the prompt to enter the first number.

Reading the first number:

mov ah, 1

int 21h

mov a, al

sub a, 48

Reads a single character from the standard input using DOS interrupt int 21h with AH=1 (read character) and stores it in variable a. It then converts the ASCII character to its equivalent numerical value by subtracting 48 ('0' in ASCII).

Calling newLine:

call newLine

Calls the newLine procedure to print a new line.

Printing "Enter the second number: ":

print "Enter the second number: "

Prints the prompt to enter the second number.

Reading the second number:

mov ah, 1

int 21h

mov b, al

sub b, 48

Reads a single character from the standard input, stores it in variable b, and converts it to its numerical value.

Subtraction:

mov ch, a

sub ch, b

add ch, 48

Subtracts the value of b from a and stores the result in ch. It then converts the result back to ASCII character by adding 48.

Printing "Subtraction: ":

print "Subtraction: "

Prints the label for the result.

Printing the result:

mov ah, 2

mov dl, ch

int 21h

Prints the result stored in ch.

Program Exit:

Exit:

mov ah, 4ch

int 21h

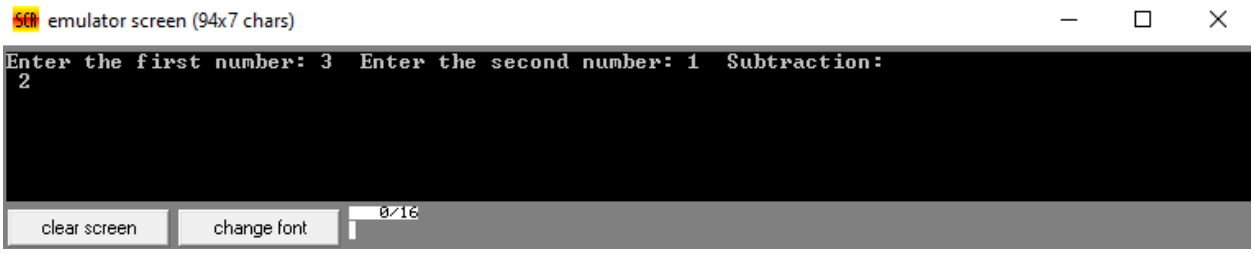
main endp

end main

Exits the program by calling DOS interrupt int 21h with AH=4Ch (terminate with return code).

Input

Output



Problem : The problem is to take three inputs. It will do the summation of the first two numbers and from that result outcome do the subtraction of the third numbers and generate the output finally.

Input: 5+2-3

Output: 4

Code

```
include "emu8086.inc"
.model small
.stack 100h

.data
    a db ?
    b db ?
    c db ?

.code
    newLine proc
        mov ah, 2
        mov dl, 10
        int 21h
        mov dl, 13
        int 21h
        ret

    main proc
        mov ax, @data
        mov ds, ax

        print "Enter the first number: "
        mov ah, 1
        int 21h
        mov a, al
        sub a, 48
       call newLine

        print "Enter the second number: "
        mov ah, 1
        int 21h
        mov b, al
        sub b, 48
        call newLine

        print "Enter the third number: "
        mov ah, 1
        int 21h
        mov c, al
        sub c, 48
        call newLine

        mov bl, a
        add bl, b
        sub bl, c

        print "The answer is: "
        mov ah, 2
        mov dl, bl
        add dl, 48
```

Code Explaination

Initialization:

include "emu8086.inc": This line includes the emu8086 library, which provides macros and constants specific to emu8086.

.model small: This directive sets the memory model to small.

.stack 100h: This directive sets the stack size to 256 bytes (100h in hexadecimal).

Data Section:

.data: This directive marks the beginning of the data section.

a db ?, b db ?, c db ?: These lines declare three uninitialized byte variables a, b, and c.

Code Section:

.code: This directive marks the beginning of the code section.

Procedure Definitions:

newLine proc: This is a procedure named newLine that prints a newline character ('\n') to the console.

mov ah, 2: Move the value 2 into the AH register, which indicates the function number for printing a character.

mov dl, 10: Move the ASCII code for the newline character (10) into the DL register.

int 21h: Call interrupt 21h (the DOS interrupt), which performs various I/O operations.

mov dl, 13: Move the ASCII code for the carriage return character (13) into the DL register.

int 21h: Call interrupt 21h to print the carriage return character.

ret: Return from the procedure.

main proc: This is the main procedure of the program.

mov ax, @data: Move the address of the data segment to the AX register.

mov ds, ax: Move the value of AX into the DS register to set up the data segment.

Input and process the first number:

print "Enter the first number: ": Display the prompt message.

mov ah, 1: Set AH to 1, indicating a keyboard input operation.

int 21h: Call interrupt 21h to read a character from the keyboard.

mov a, al: Move the input character (ASCII code) to the variable a.

sub a, 48: Convert the ASCII digit to its numeric value by subtracting 48.

call newLine: Call the newLine procedure to print a newline.

Input and process the second number (similar to the first number).

Input and process the third number (similar to the first number).

Calculate the result:

mov bl, a: Move the value of a to the BL register.

add bl, b: Add the value of b to BL.

sub bl, c: Subtract the value of c from BL.

Output the result:

print "The answer is: ": Display the result prompt.

mov ah, 2: Set AH to 2, indicating a display character operation.

mov dl, bl: Move the result value in BL to DL.

add dl, 48: Convert the numeric result back to ASCII by adding 48.

int 21h: Call interrupt 21h to print the result.

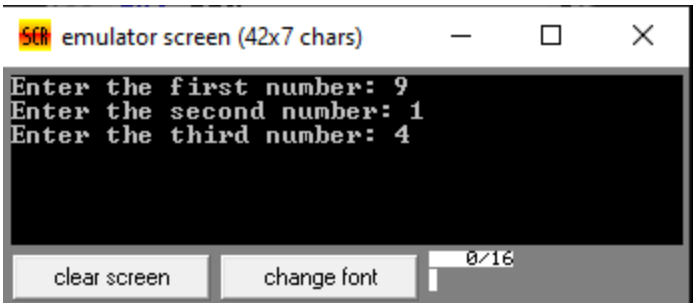
Exit: Label indicating the end of the program.

mov ah, 4ch: Set AH to 4Ch, indicating the exit function.

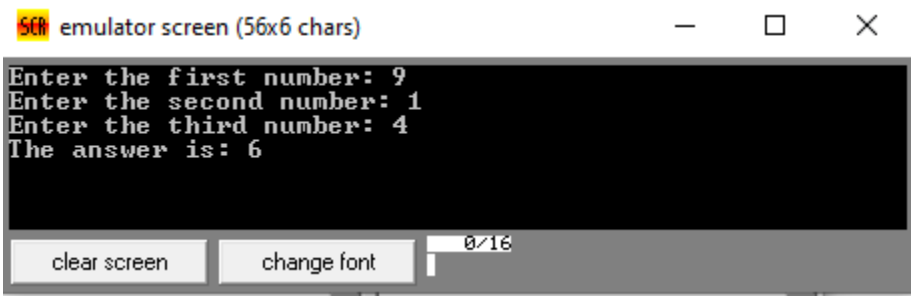
int 21h: Call interrupt 21h to exit the program.

end main: End of the main procedure.

Input


emulator screen (42x7 chars)
Enter the first number: 9
Enter the second number: 1
Enter the third number: 4
clear screen
change font
0/16

Output


emulator screen (56x6 chars)
Enter the first number: 9
Enter the second number: 1
Enter the third number: 4
The answer is: 6
clear screen
change font
0/16