Lab # 01
Introduction to Assembly and Assembler

OBJECTIVE
Developing basic understanding about Assembly.

INTRODUCTION

Programming Languages
A programming language is an artificial language that can be used to control the behavior of a machine, particularly a computer. Programming languages, like human languages, have syntactic and semantic rules to define meaning**.**

Types of Programming Languages
Programming languages can be classified into three basic categories on the basis of understanding level of users as well as the machine to which instructions has been given:

1. High Level Languages
A programming language that enables a programmer to write programs that are more or less independent of a particular type of computer and are designed to give a better program efficiency. Such languages are considered high-level because they are closer to human languages.

2. Low Level Languages
These are designed to have both: a relatively good programming efficiency and relatively good machine efficiency.

3. Machine Language
Machine language is at the lowest level, because it is the actual binary code of 1s and 0s that the computer understands. These are designed to give a better machine efficiency.

Registers Classification
The registers inside the microprocessor are classified according to the function they perform In general, they are classified as
1. Data registers
2. Address registers
3. Segment register
4. Offset registers
5. Status register

Some General Purpose Registers:

AX (Accumulator Register)
- It is the preferred register to use in the arithmetic, logic and data transfer instructions because its use generates the shortest machine code.
- In multiplication and division operations, one of the numbers involved must be in AX or AL.
- Input and output operation also requires the use of AX and AL

BX (Base Register)

- It is used to store the data also it serves as an address register.

CX (Count Register)

- Program loop instructions are facilitated by the use of CX register, serves as a loop counter.
- Also used as a counter in the string operations.
- CL is used as count in instructions that shift and rotate bits.

DX (Data Register)

- It is used in multiplication and division operations.
- It is used in IO operation like DL in character output and DX in string output functions.

Register Size:

- We have three different sizes of registers:
- 8-bit register: AH, AL, BH, BL, CH, CL, DH, DL
- 16-bit registers: AX, BX, CX, DX, SP, BP, SI, DI, SS, DS, CS, ES, FS, GS, IP, FLAGS
- 32-bit registers: EAX, EXB, ECX, EDX, ESI, EDI, ESP, EBP, EIP, and EFLAGS.

Basic MOV Instruction

- The basic MOV instruction is used to transfer data between registers, between and memory locations, or to have a number directly to a register or memory location.

Syntax: MOV Destination, Source

Examples:

- MOV AH, BL ; 8-bits register to register
- MOV BX, AX ; 16-bits register to register
- MOV byte1, BL ; 8-bit register to memory

MOV AX, word1 ;16-bit memory to register

Code to Display String

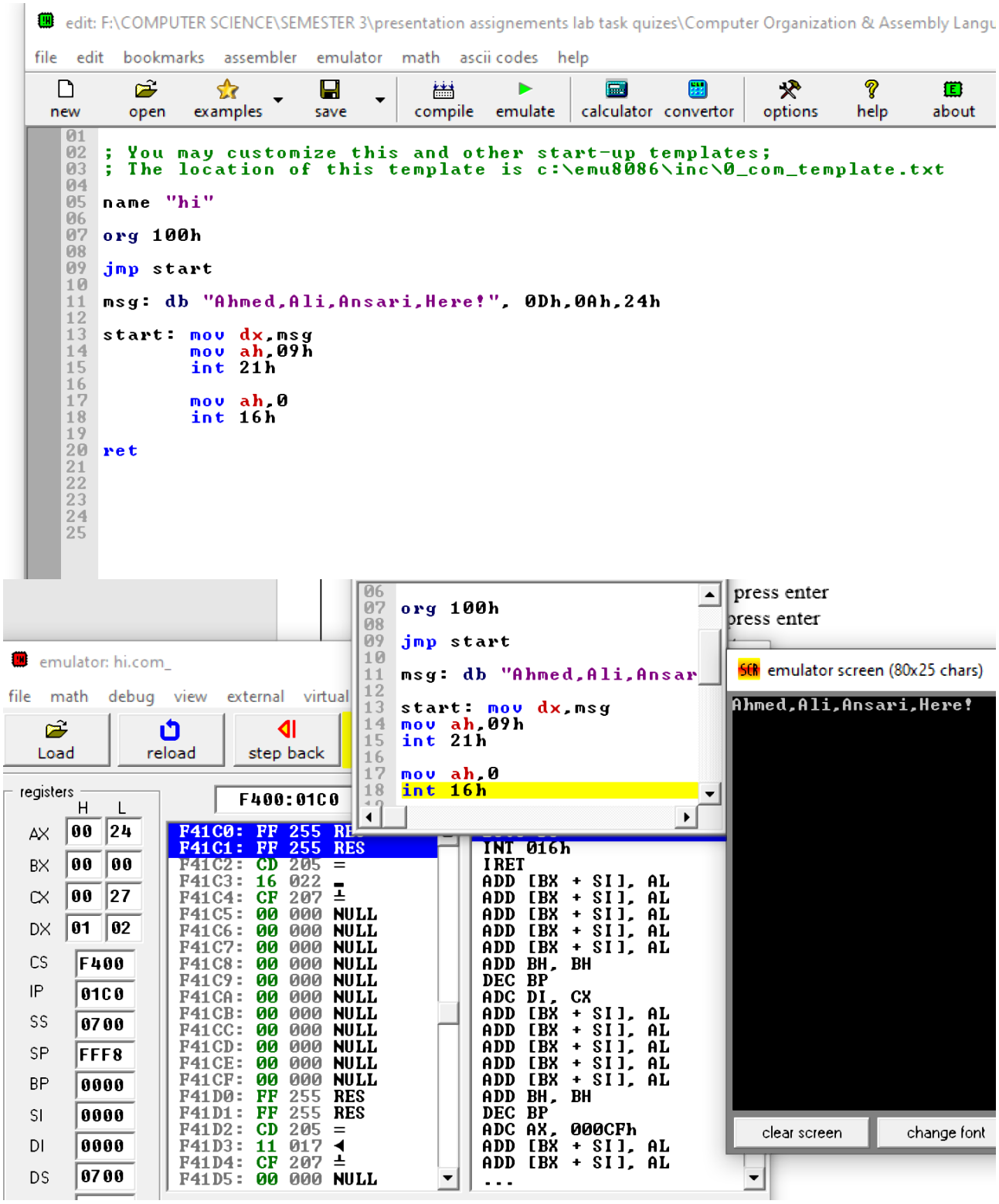
```
.MODEL SMALL
.STACK 100H
.DATA
        MESSAGE1 DB 0AH, 0DH, "INDUS UNIVERSITY$"
.CODE
MAIN:
        MOV AX, @DATA
        MOV DS, AX
        MOV DX, OFFSET MESSAGE1
        MOV AH, 09H
        INT 21H

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

Procedure

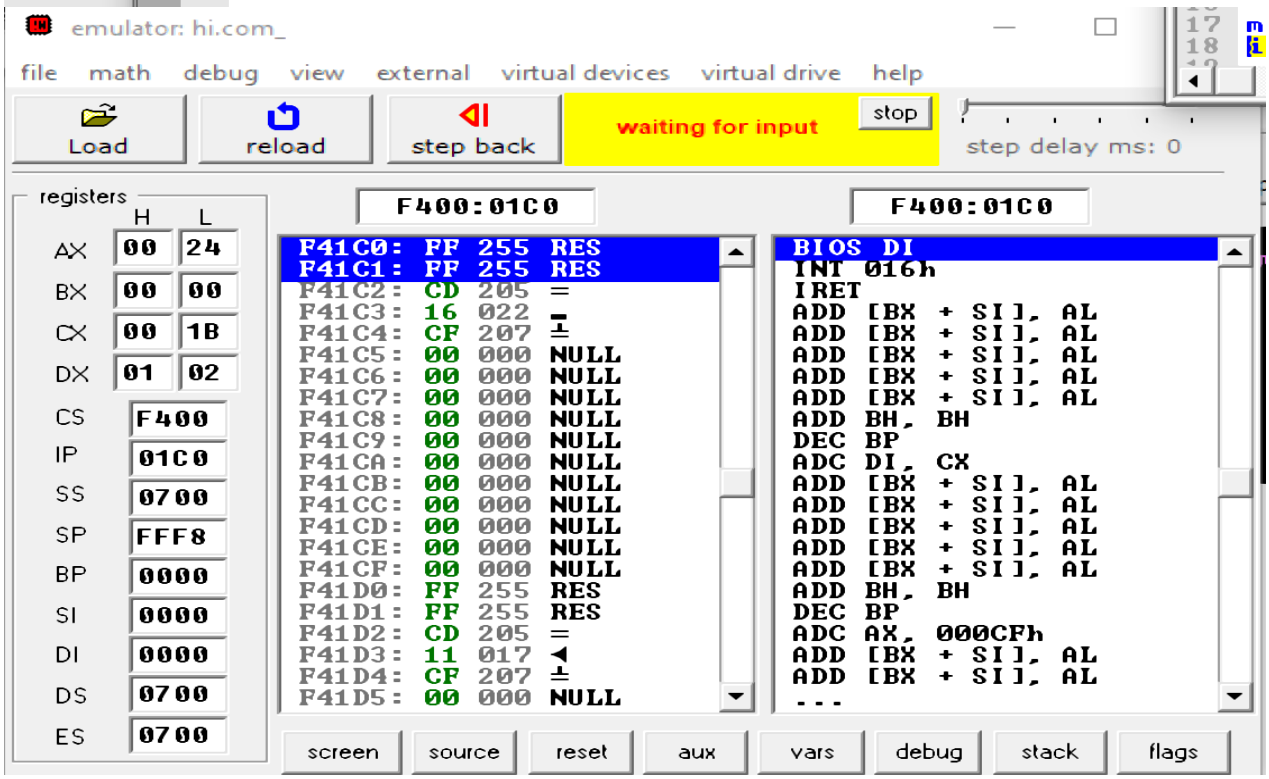
- Open a notepad editor and create a new file with any name but with the extension ".asm", eg lab1.asm. save it to bin folder of tasm.
- Now copy and paste the following code into that file and save again
- Goto command prompt and type cd c:\tasm\bin
- Now type "tasm filename.asm" and press enter
- Now type "tlink filename.obj" and press enter
- Now type "filename.exe" and hit enter

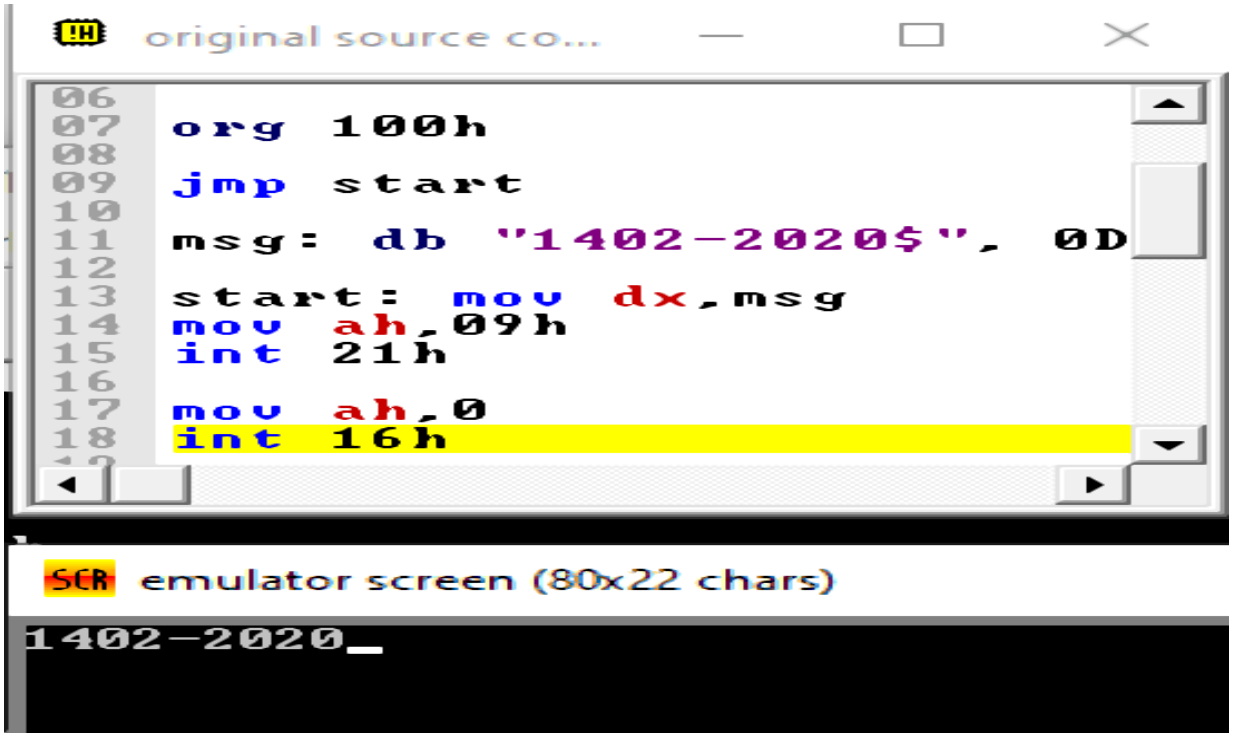
Lab Tasks

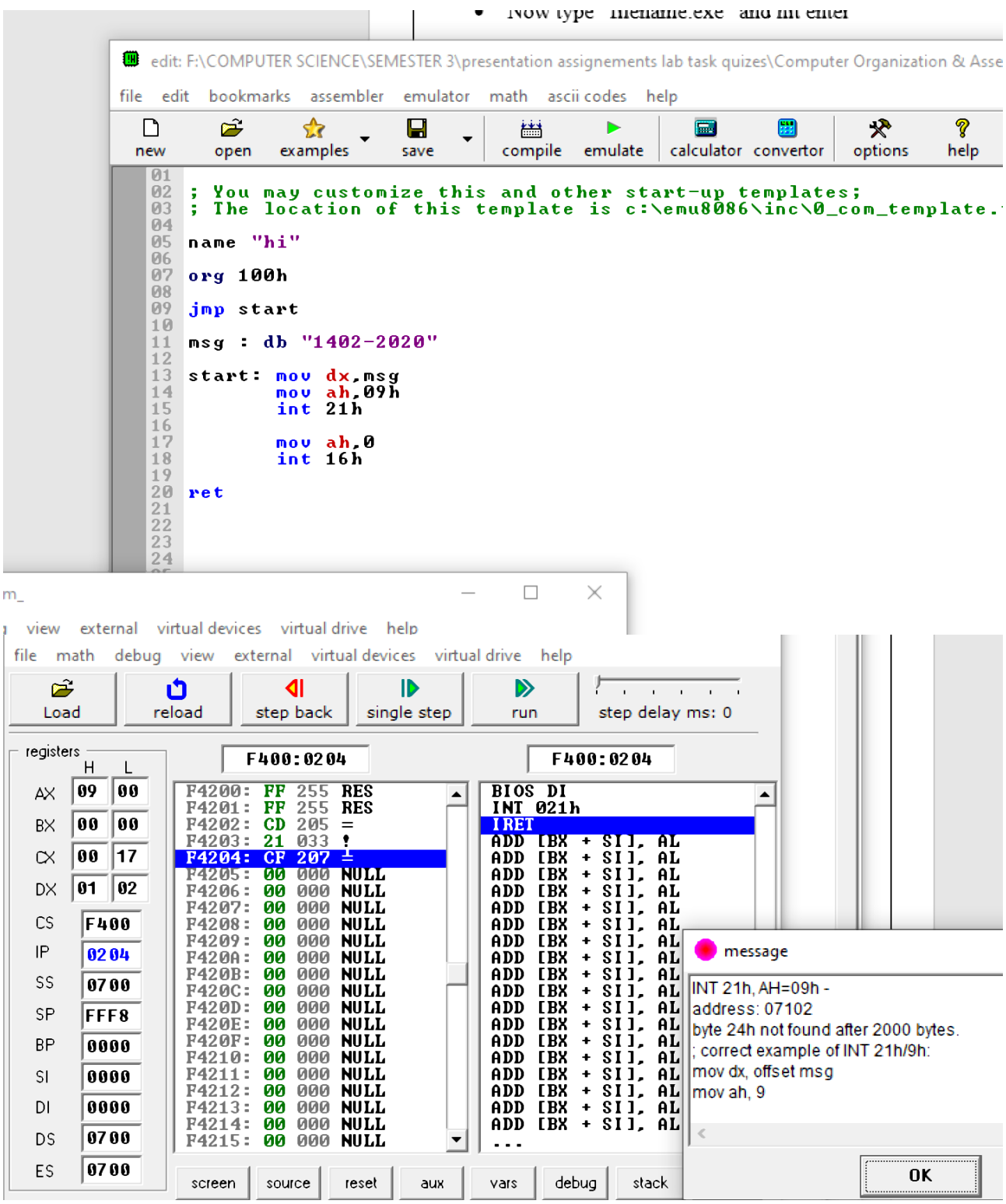
1. Write a program to display your complete name on the screen
2. Write a program to display you complete enrolment number on screen
3. Test your code after removing the $ sign from the string


edit: F:\COMPUTER SCIENCE\SEMESTER 3\presentation assignements lab task quizes\Computer Organization & Assembly Langu
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt
name "hi"
org 100h
jmp start
msg: db "Ahmed,Ali,Ansari,Here!", 0Dh,0Ah,24h
start: mov dx,msg
mov ah,09h
int 21h
mov ah,0
int 16h
ret
emulator: hi.com_
file math debug view external virtual
Load reload step back
registers
H L
AX 00 24
BX 00 00
CX 00 27
DX 01 02
CS F400
IP 01C0
SS 0700
SP FFF8
BP 0000
SI 0000
DI 0000
DS 0700
F400:01C0
F41C0: FF 255 RES
F41C1: FF 255 RES
F41C2: CD 205 =
F41C3: 16 022
F41C4: CF 207
F41C5: 00 000 NULL
F41C6: 00 000 NULL
F41C7: 00 000 NULL
F41C8: 00 000 NULL
F41C9: 00 000 NULL
F41CA: 00 000 NULL
F41CB: 00 000 NULL
F41CC: 00 000 NULL
F41CD: 00 000 NULL
F41CE: 00 000 NULL
F41CF: 00 000 NULL
F41D0: FF 255 RES
F41D1: FF 255 RES
F41D2: CD 205 =
F41D3: 11 017
F41D4: CF 207
F41D5: 00 000 NULL
INT 016h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC DI, CX
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC AX, 000CFh
ADD [BX + SI], AL
ADD [BX + SI], AL
...
org 100h
jmp start
msg: db "Ahmed,Ali,Ansar
start: mov dx,msg
mov ah,09h
int 21h
mov ah,0
int 16h
press enter
press enter
emulator screen (80x25 chars)
Ahmed,Ali,Ansari,Here!
clear screen
change font

```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.t

name "hi"

org 100h

jmp start

msg: db "1402-2020$", 0Dh,0Ah,24h

start: mov dx,msg
       mov ah,09h
       int 21h

       mov ah,0
       int 16h

ret
```




original source co...
org 100h
jmp start
msg: db "1402-2020$", 0D
start: mov dx,msg
mov ah,09h
int 21h
mov ah,0
int 16h
emulator screen (80x22 chars)
1402-2020


edit: F:\COMPUTER SCIENCE\SEMESTER 3\presentation assignements lab task quizes\Computer Organization & Asse
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.
name "hi"
org 100h
jmp start
msg : db "1402-2020"
start: mov dx,msg
mov ah,09h
int 21h
mov ah,0
int 16h
ret
view external virtual devices virtual drive help
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
AX 09 00
BX 00 00
CX 00 17
DX 01 02
CS F400
IP 0204
SS 0700
SP FFF8
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ±
F400:0204
BIOS DI
INT 021h
IRET
message
INT 21h, AH=09h -
address: 07102
byte 24h not found after 2000 bytes.
; correct example of INT 21h/9h:
mov dx, offset msg
mov ah, 9
OK
screen source reset aux vars debug stack