Computer Architecture and Assembly Language [CSIT131] BCA 2nd Semester
Practical End Semester Exam Date 08-07-2021

By: Esha_Suchdeo

Roll No: A7304820015

Question 1 is compulsory Attempt any one from the remaining two, write output and comments for readability: Roll Nos 3,6,9,12,15,18,21,24,27,30

Ques 1. Write an assembly code to find factorial of a given Number and highlight the results in the registers/LED.
Ques 2. To develop an Assembly Code to multiply two Hexadecimal numbers. display the result at the LED.
Ques 3. To develop an Assembly Code to initialize variables in the memory.

Assembly program to calculate the factorial of a number inputed by user.

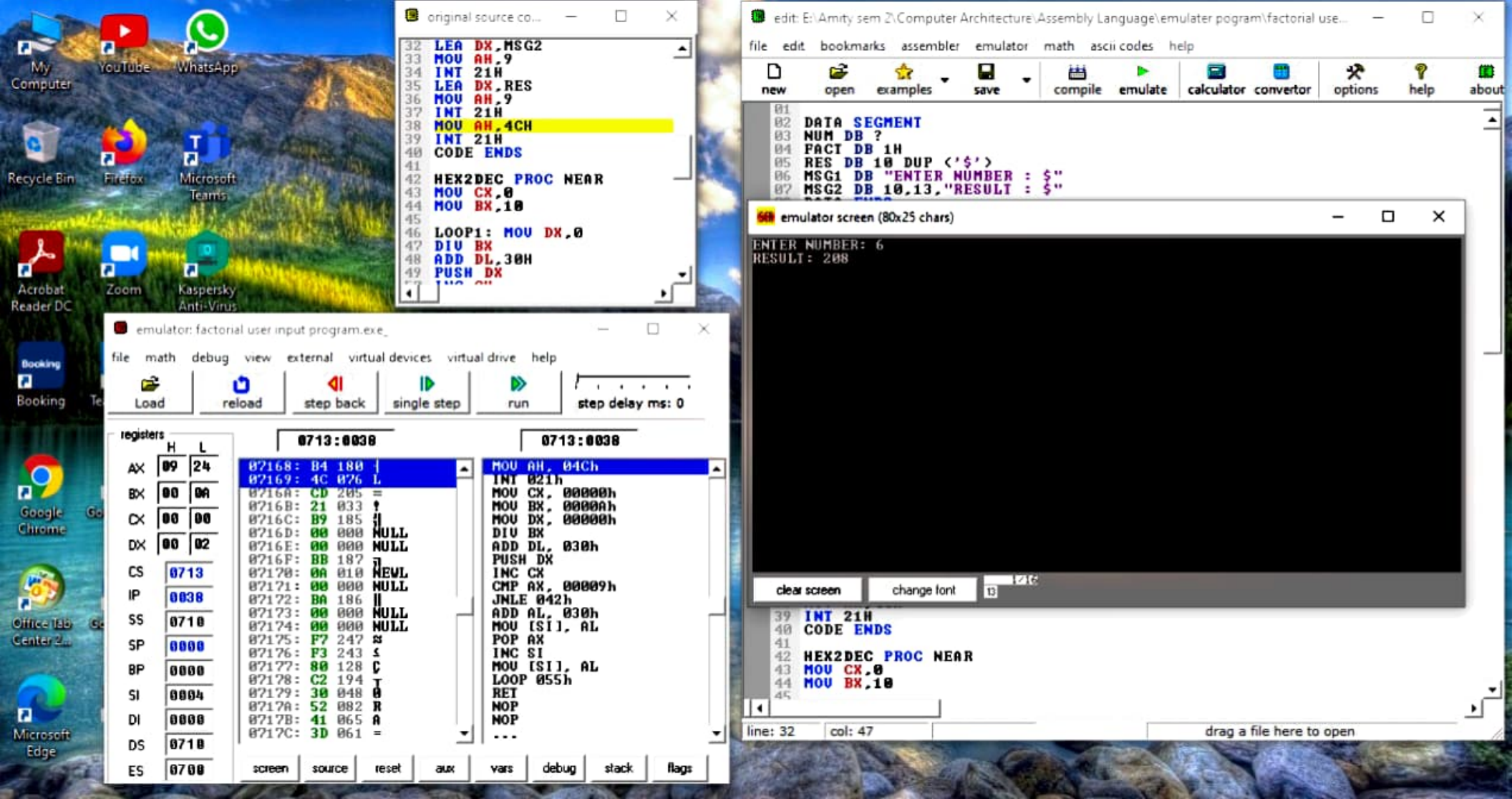
```
data segment ; variable declaration
num db ?
fact db 1H
res db 10 dup('$') ; array of size 10
msg1 db "Enter number : $"
msg2 db 10, 13, "Result : $"

Data ends

Code segment
assume ds : data, CS : Code
start:
mov ax, data } ; data is transfered
mov ds, ax }   to ds register
lea dx, MSG1 } ; A number is
mov ah, 9      inputted using
int 21H        message & interrupts
mov ah, 1
int 21H
sub al, 30H ; 30 is sub from no.
mov num, al ; no is transfered to al
mov ah, 0 ; ah is set as 0
mov al, Fact ; fact's value is put
mov ch, 0 ; ch is set as 0
```

int al

```
mov cl, num
Label 1: Mul cl          ; Loop
loop label 1
lea si, res
call hex2dec
lea dx, msg2
mov ah, 9
int 21H
lea dx, res
mov ah, 9
int 21H
mov ah, 4ch
int 21H
code ends
hex2dec proc near ; function start
mov Cx, 0
mov bx, 10
Loop 1: mov dx, 0
div Bx
add dl, 30h
push dl, 30h
inc Cx
Cmp ax, 9
Jg loop 1
add al, 30h
mov[SI], Al          ; loop body
loop2: POP ax
inc SI
mov SI, al
```

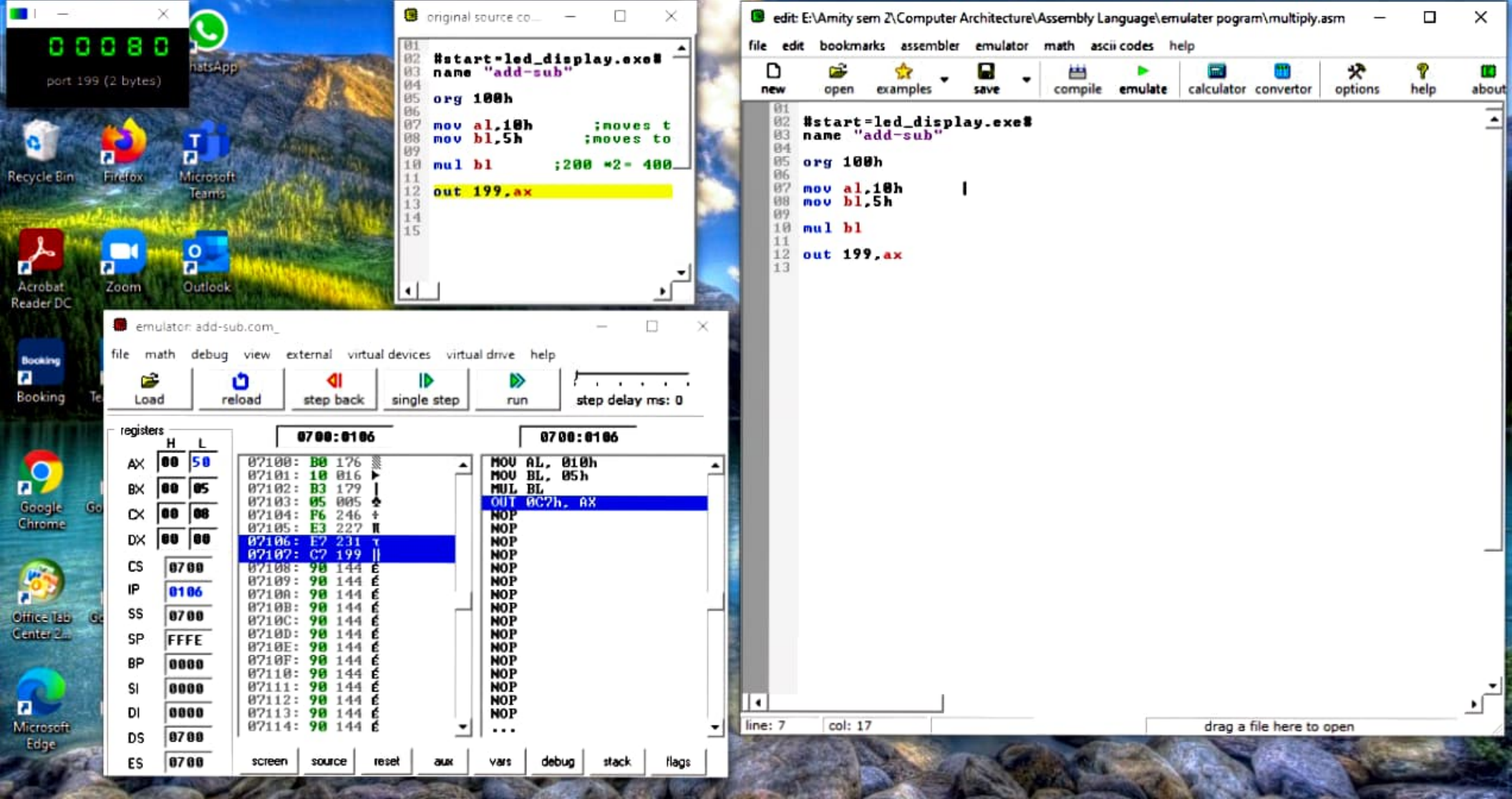
```
loop loop2 ; end of loop
RET        ; return statement
Hex2dec Endp ; return of prog
End start    ; end of start
```

- Output : input = 6 ; output = 208


original source co...
LEA DX,MSG2
MOV AH,9
INT 21H
LEA DX,RES
MOV AH,9
INT 21H
MOV AH,4CH
INT 21H
CODE ENDS
HEX2DEC PROC NEAR
MOV CX,0
MOV BX,10
LOOP1: MOV DX,0
DIV BX
ADD DL,30H
PUSH DX
edit: E:\Amity sem 2\Computer Architecture\Assembly Language\emulater pogram\factorial use...
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
DATA SEGMENT
NUM DB ?
FACT DB 1H
RES DB 10 DUP ('$')
MSG1 DB "ENTER NUMBER : $"
MSG2 DB 10,13,"RESULT : $"
emulator screen (80x25 chars)
ENTER NUMBER: 6
RESULT: 208
clear screen
change font
INT 21H
CODE ENDS
HEX2DEC PROC NEAR
MOV CX,0
MOV BX,10
line: 32
col: 47
drag a file here to open
emulator: factorial user input program.exe
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 09 24
BX 00 0A
CX 00 00
DX 00 02
CS 0713
IP 0038
SS 0710
SP 0000
BP 0000
SI 0004
DI 0000
DS 0710
ES 0700
0713:0038
07168: B4 180
07169: 4C 076 L
0716A: CD 205
0716B: 21 033 !
0716C: B9 185
0716D: 00 000 NULL
0716E: 00 000 NULL
0716F: BB 187
07170: 0A 010 NEWL
07171: 00 000 NULL
07172: BA 186
07173: 00 000 NULL
07174: 00 000 NULL
07175: F7 247
07176: F3 243
07177: 80 128
07178: C2 194
07179: 30 048 0
0717A: 52 082 R
0717B: 41 065 A
0717C: 3D 061 =
0713:0038
MOV AH, 04Ch
INT 021h
MOV CX, 00000h
MOV BX, 0000Ah
MOV DX, 00000h
DIV BX
ADD DL, 030h
PUSH DX
INC CX
CMP AX, 00009h
JNLE 042h
ADD AL, 030h
MOV [SI], AL
POP AX
INC SI
MOV [SI], AL
LOOP 055h
RET
NOP
NOP
screen source reset aux vars debug stack flags

Q2 To develop an Assembly Code to multiply two Hexadecimal numbers display the result at the LED.

```
#start = led-display.exe#
name "hex-mul"

org 100h

mov al, 10h        ; putting 10h to al
mov bl, 5h         ; putting 5 to bl

mul bl             ; multiplying bl with

out 199, ax        ; statement to
                   show output in LED
```


00080
port 199 (2 bytes)
original source co...
#start=led_display.exe#
name "add-sub"
org 100h
mov al,10h ;moves t
mov bl,5h ;moves to
mul bl ;200 *2= 400
out 199,ax
edit: E:\Amity sem 2\Computer Architecture\Assembly Language\emulater program\multiply.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
#start=led_display.exe#
name "add-sub"
org 100h
mov al,10h
mov bl,5h
mul bl
out 199,ax
line: 7 col: 17
drag a file here to open
emulator: add-sub.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 50
BX 00 05
CX 00 08
DX 00 00
CS 0700
IP 0106
SS 0700
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
0700:0106
0700:0106
07100: B0 176
07101: 10 016
07102: B3 179
07103: 05 005
07104: F6 246
07105: E3 227
07106: E7 231
07107: C7 199
07108: 90 144
07109: 90 144
0710A: 90 144
0710B: 90 144
0710C: 90 144
0710D: 90 144
0710E: 90 144
0710F: 90 144
07110: 90 144
07111: 90 144
07112: 90 144
07113: 90 144
07114: 90 144
MOV AL, 010h
MOV BL, 05h
MUL BL
OUT 0C7h, AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags