CS1107:

COMPUTER ARCHITECTURE

LAB FILE

CODES

SUBMITTED BY:

KONDROLLA DINESH REDDY

(2020BTechCSE040)

FACULTY GUIDE:

MR. DIVANSHU JAIN

Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

FEB 2022

EXPERIMENT – 1:

PART-A:

AIM: WAP for 8-bit Addition

THEORY:

As we have taken 2 hexadecimal number as follows:

0 F F H

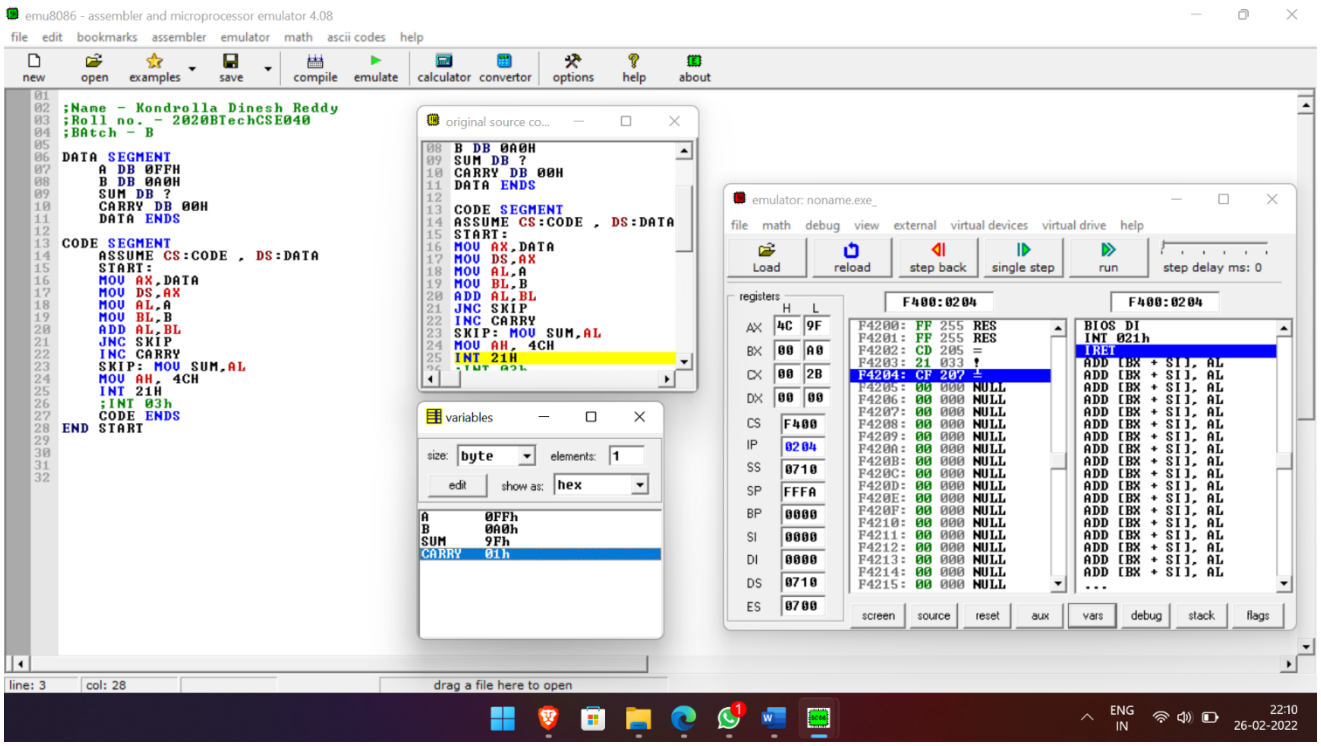
0 A 0 H

---

1 0 9 F H

It will give a carry of 1, so our answer will be 0 9 F H and carry will be 1.

On emulating the code we'll get the following –



```
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;BAtch - B

DATA SEGMENT
    A DB 0FFH
    B DB 0A0H
    SUM DB ?
    CARRY DB 00H
    DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE , DS:DATA
    START:
    MOV AX,DATA
    MOV DS,AX
    MOV AL,A
    MOV BL,B
    ADD AL,BL
    JNC SKIP
    INC CARRY
    SKIP: MOV SUM,AL
    MOV AH, 4CH
    INT 21H
    ;INT 03h
    CODE ENDS
END START
```

PART-B:

AIM: WAP for 16-bit Addition

THEORY:

As we have taken 2 hexadecimal number as follows:

0 F F A 0 H

0 A A 1 0 H

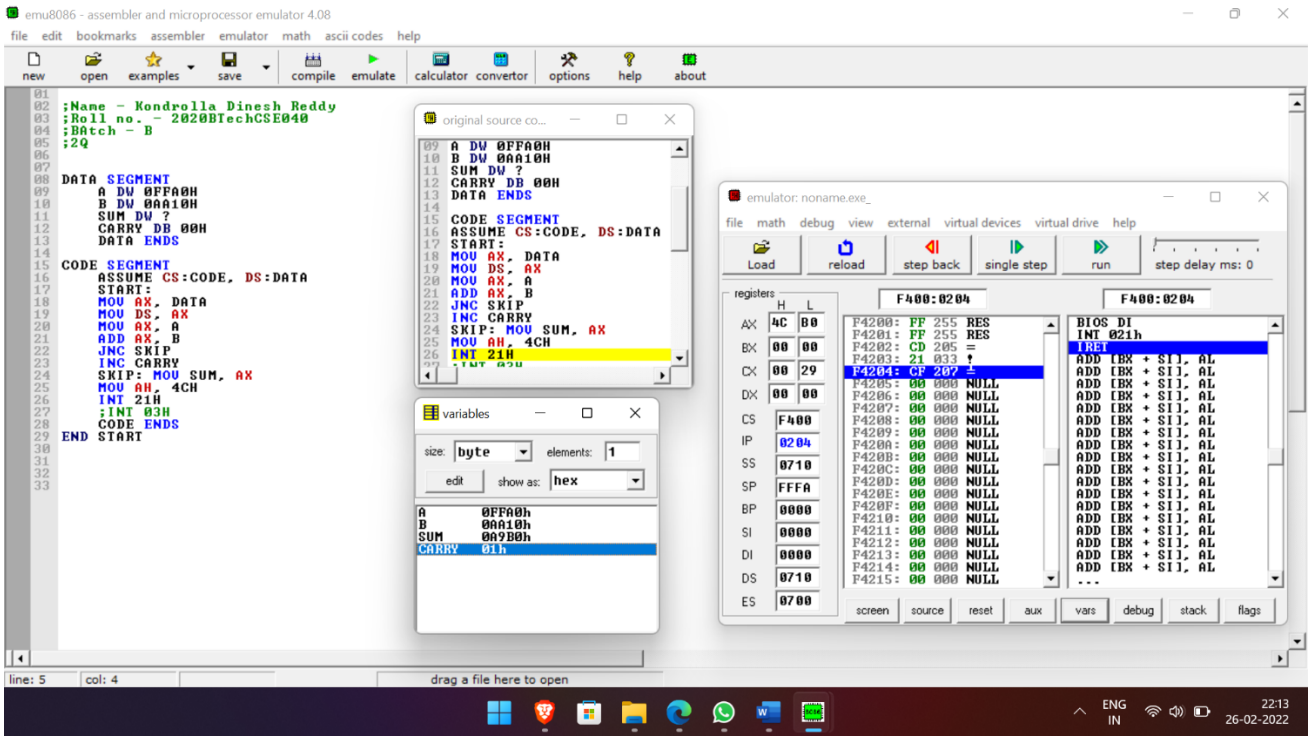
---

1 0 A 9 B 0 H

It will give a carry of 1, so our answer will be 0 A 9 B 0 H and carry will be 1.

On emulating the code we'll get the following –

```
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;BAtch - B
;2Q


DATA SEGMENT
     A DW 0FFA0H
     B DW 0AA10H
     SUM DW ?
     CARRY DB 00H
     DATA ENDS

CODE SEGMENT
     ASSUME CS:CODE, DS:DATA
     START:
     MOV AX, DATA
     MOV DS, AX
     MOV AX, A
     ADD AX, B
     JNC SKIP
     INC CARRY
     SKIP: MOV SUM, AX
     MOV AH, 4CH
     INT 21H
     ;INT 03H
     CODE ENDS
END START
```


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;BAtch - B
;2Q
DATA SEGMENT
A DW 0FFA0H
B DW 0AA10H
SUM DW ?
CARRY DB 00H
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
MOV AX, DATA
MOV DS, AX
MOV AX, A
ADD AX, B
JNC SKIP
INC CARRY
SKIP: MOV SUM, AX
MOV AH, 4CH
INT 21H
;INT 03H
CODE ENDS
END START
original source co...
variables
size: byte elements: 1
edit show as: hex
A 0FFA0h
B 0AA10h
SUM 0A9B0h
CARRY 01h
emulator: noname.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
AX 4C B0
BX 00 00
CX 00 29
DX 00 00
CS F400
IP 0204
SS 0710
SP FFFA
BP 0000
SI 0000
DI 0000
DS 0710
ES 0700
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ±
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
F400:0204
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
screen source reset aux vars debug stack flags
line: 5 col: 4
drag a file here to open
ENG IN
22:13
26-02-2022

EXPERIMENT – 2:

PART-A:

AIM: WAP for 8-bit Unsigned Subtraction

THEORY:

As we have taken 2 hexadecimal number as follows:

0 3 6 H

0 4 2 H

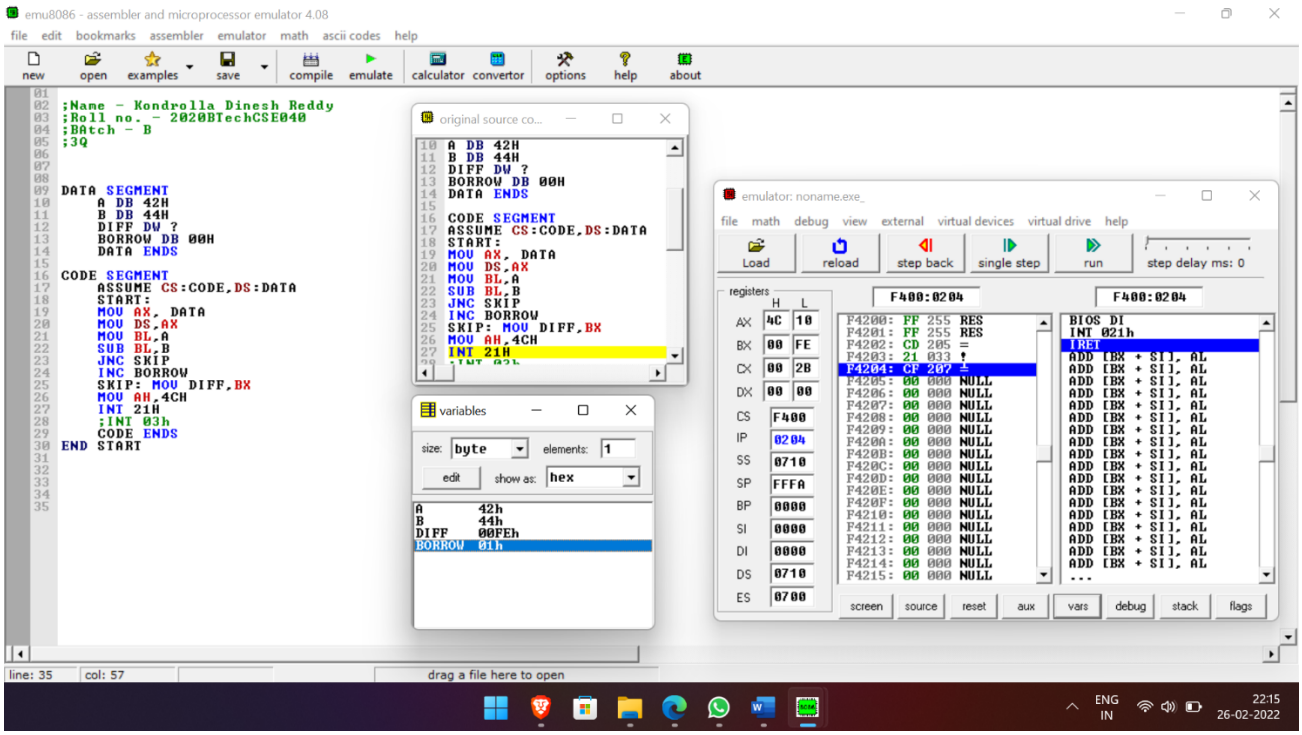
---

1 0 F 4 H

It will take a borrow of 1, so our answer will be 0 F 4 H and borrow will be 1.

On emulating the code we'll get the following –

```
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;BAtch - B
;3Q


DATA SEGMENT
    A DB 42H
    B DB 44H
    DIFF DW ?
    BORROW DB 00H
    DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    START:
    MOV AX, DATA
    MOV DS,AX
    MOV BL,A
    SUB BL,B
    JNC SKIP
    INC BORROW
    SKIP: MOV DIFF,BX
    MOV AH,4CH
    INT 21H
    ;INT 03h
    CODE ENDS
END START
```


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;BAtch - B
;3Q
DATA SEGMENT
A DB 42H
B DB 44H
DIFF DW ?
BORROW DB 00H
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX, DATA
MOV DS,AX
MOV BL,A
SUB BL,B
JNC SKIP
INC BORROW
SKIP: MOV DIFF,BX
MOV AH,4CH
INT 21H
;INT 03h
CODE ENDS
END START
original source co...
variables
size: byte elements: 1
edit show as: hex
A 42h
B 44h
DIFF 00FEh
BORROW 01h
emulator: noname.exe_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
F400:0204
F400:0204
BIOS DI
INT 021h
IRET
screen source reset aux vars debug stack flags
line: 35 col: 57
drag a file here to open
ENG IN
22:15
26-02-2022

PART-B

AIM: WAP for 16-bit Unsigned Subtraction

THEORY:

As we have taken 2 hexadecimal number as follows:

0 F F A 0 H

0 A A 1 0 H

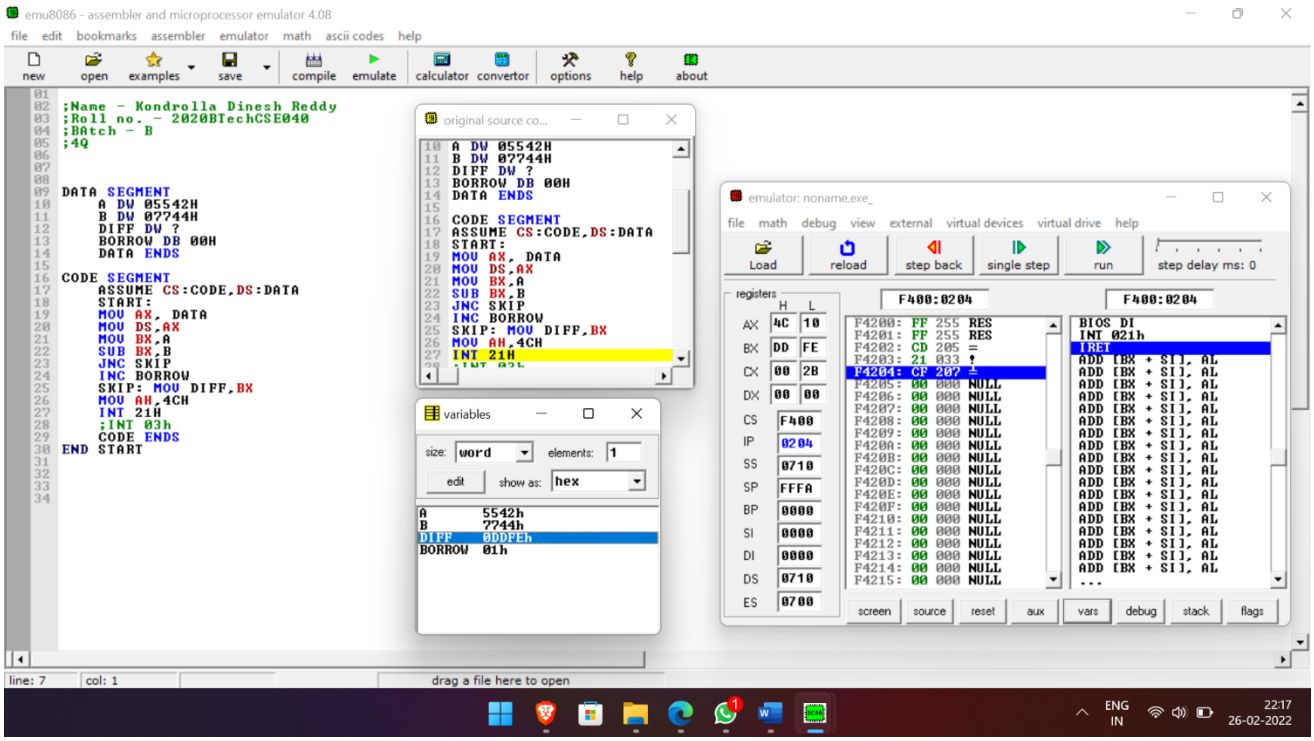
---

0 5 5 9 0 H

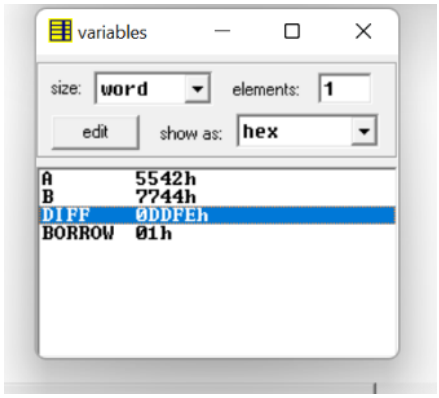
It will take a borrow of 1, so our answer will be 0 5 5 9 0 H and borrow will be 0.

On emulating the code we'll get the following –

```
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;BAtch - B
;4Q


DATA SEGMENT
    A DW 05542H
    B DW 07744H
    DIFF DW ?
    BORROW DB 00H
    DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    START:
    MOV AX, DATA
    MOV DS,AX
    MOV BX,A
    SUB BX,B
    JNC SKIP
    INC BORROW
    SKIP: MOV DIFF,BX
    MOV AH,4CH
    INT 21H
    ;INT 03h
    CODE ENDS
END START
```


emu8086 - assembler and microprocessor emulator 4.08
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;BAtch - B
;4Q
DATA SEGMENT
A DW 05542H
B DW 07744H
DIFF DW ?
BORROW DB 00H
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX, DATA
MOV DS,AX
MOV BX,A
SUB BX,B
JNC SKIP
INC BORROW
SKIP: MOV DIFF,BX
MOV AH,4CH
INT 21H
;INT 03h
CODE ENDS
END START
variables
size: word
elements: 1
edit
show as: hex
A 5542h
B 7744h
DIFF 0DDFEh
BORROW 01h
emulator: noname.exe_
Load
reload
step back
single step
run
step delay ms: 0
F400:0204
BIOS DI
INT 021h
IRET

RESULT:


variables
size: word
elements: 1
edit
show as: hex
A 5542h
B 7744h
DIFF 0DDFEh
BORROW 01h

EXPERIMENT – 3:

AIM: WAP for 16-bit Unsigned Subtraction

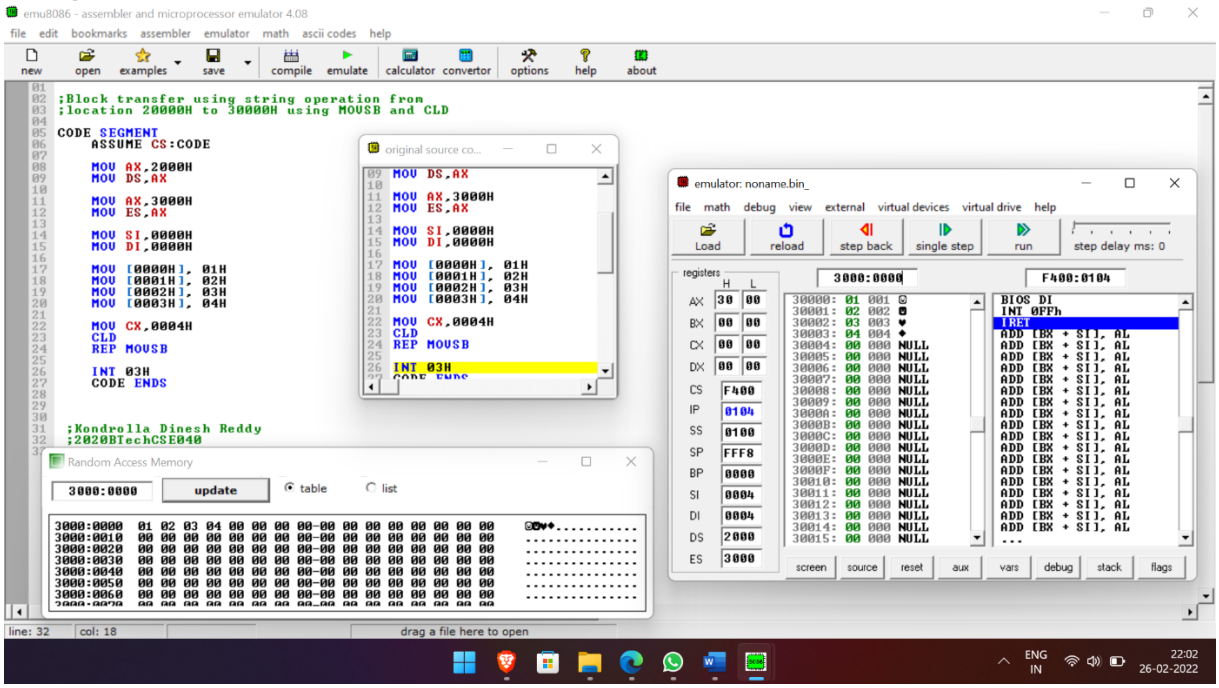
THEORY:

On emulating and running the code, we get the following windows and after we'll click the button named memory in aux to check whether the block of data has been transferred to the destination or not by checking the data of it using address.

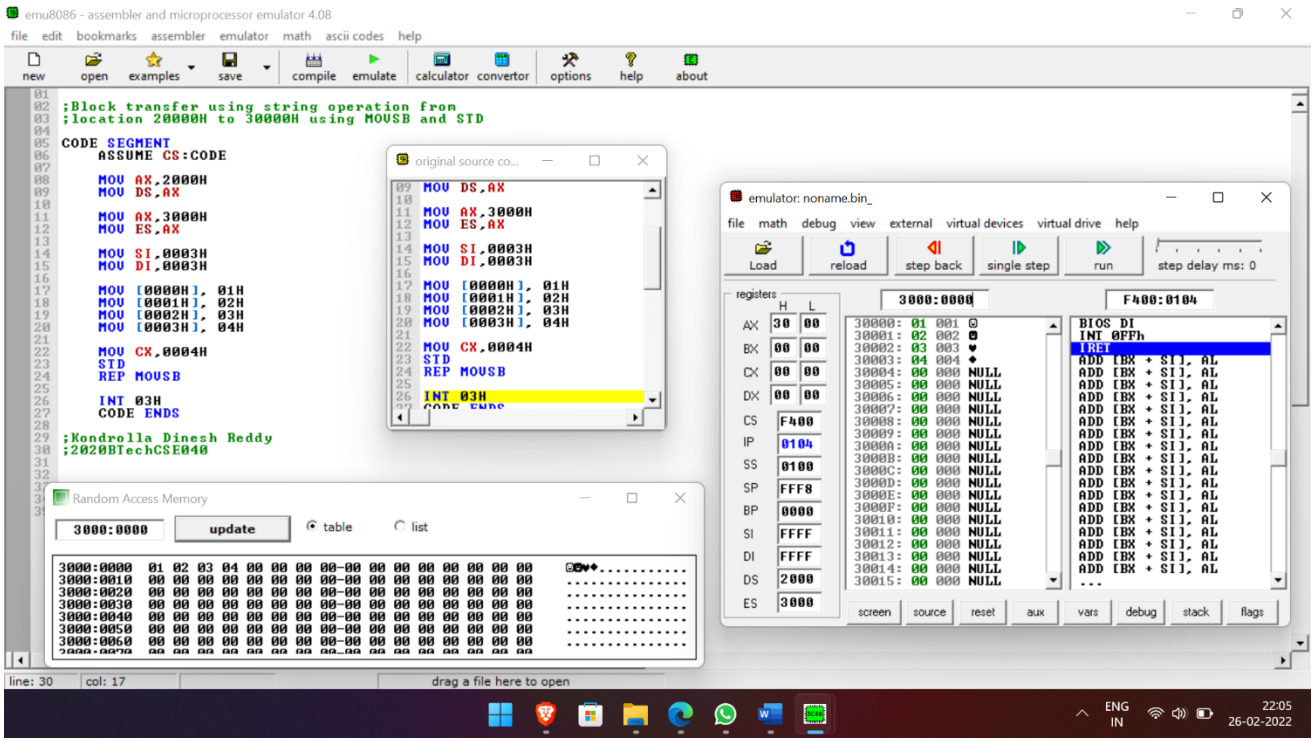
For the whole data transfer the data should be there in the source (2000) and destination (3000).

CODE:

- Block transfer using string operation from location 20000H to 30000H using MOVSB



- Block transfer using string operation from location 20000H to 30000H using MOVSB and STD.



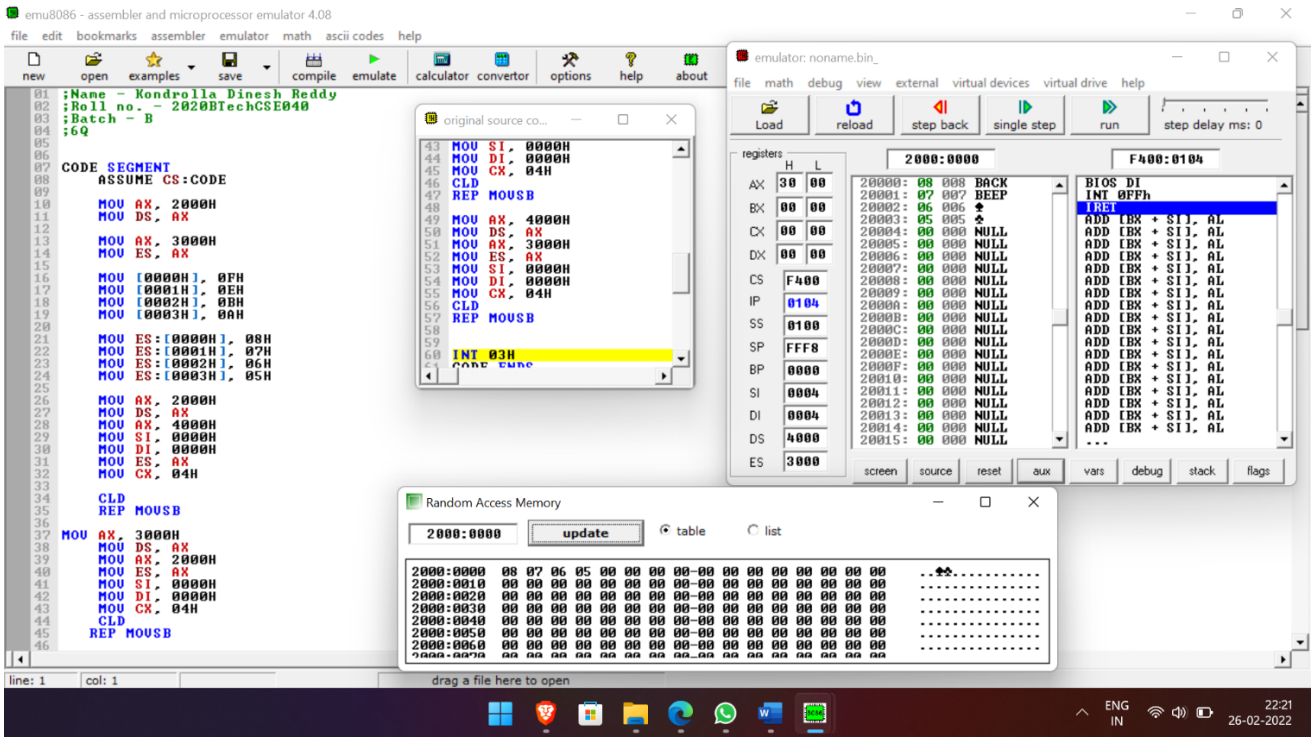
EXPERIMENT – 4:

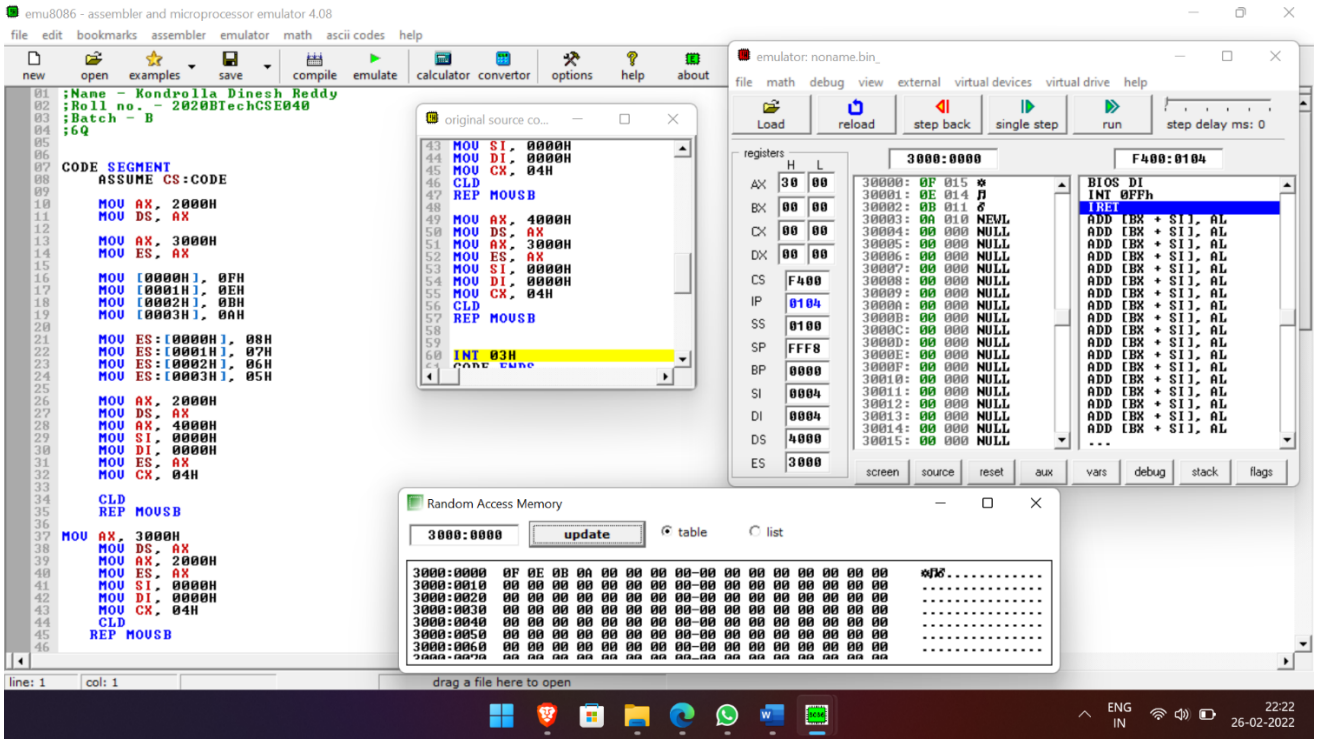
AIM: Block Exchange using String Operation

THEORY:

Block exchange program has exchanged the data of 20000H and 30000H which means that the data of 20000H has been stored in the 30000H and the data of the 30000H has been stored to 20000H.

```
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;Batch - B
;6Q


CODE SEGMENT
    ASSUME CS:CODE

    MOV AX, 2000H
    MOV DS, AX

    MOV AX, 3000H
    MOV ES, AX

    MOV [0000H], 0FH
    MOV [0001H], 0EH
    MOV [0002H], 0BH
    MOV [0003H], 0AH

    MOV ES:[0000H], 08H
    MOV ES:[0001H], 07H
    MOV ES:[0002H], 06H
    MOV ES:[0003H], 05H

    MOV AX, 2000H
    MOV DS, AX
    MOV AX, 4000H
    MOV SI, 0000H
    MOV DI, 0000H
    MOV ES, AX
    MOV CX, 04H

    CLD
    REP MOVSB

MOV AX, 3000H
    MOV DS, AX
    MOV AX, 2000H
    MOV ES, AX
    MOV SI, 0000H
    MOV DI, 0000H
    MOV CX, 04H
    CLD
   REP MOVSB

```

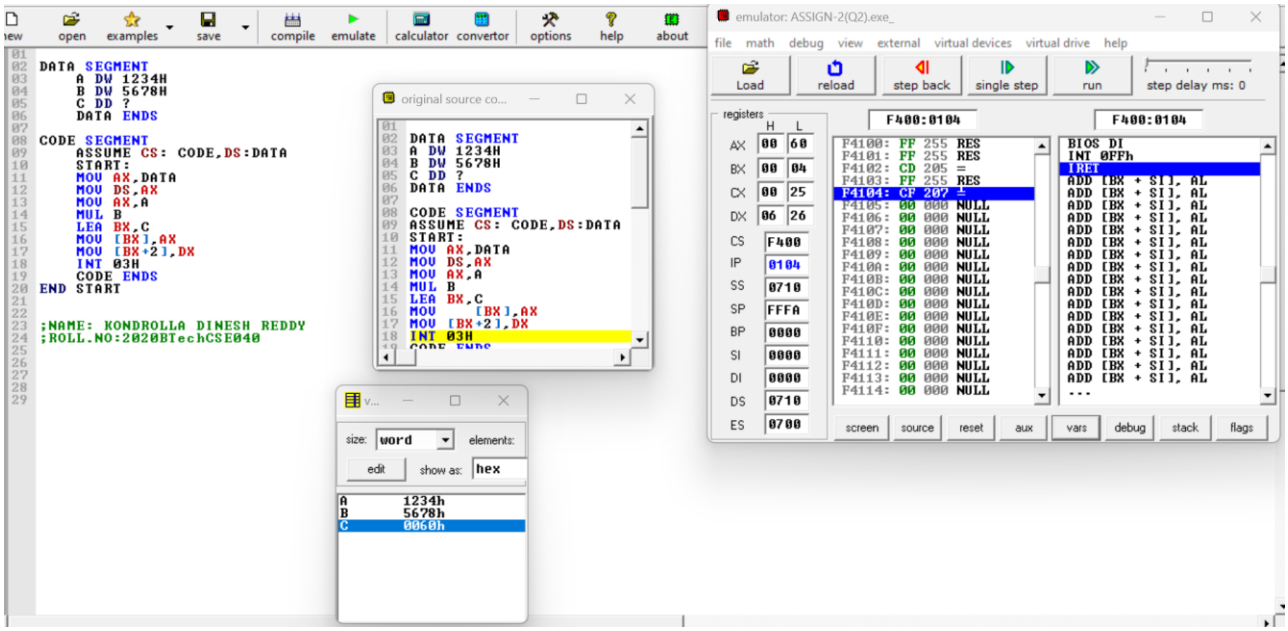

emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;Batch - B
;6Q
CODE SEGMENT
ASSUME CS:CODE
MOV AX, 2000H
MOV DS, AX
MOV AX, 3000H
MOV ES, AX
MOV [0000H], 0FH
MOV [0001H], 0EH
MOV [0002H], 0BH
MOV [0003H], 0AH
MOV ES:[0000H], 08H
MOV ES:[0001H], 07H
MOV ES:[0002H], 06H
MOV ES:[0003H], 05H
MOV AX, 2000H
MOV DS, AX
MOV AX, 4000H
MOV SI, 0000H
MOV DI, 0000H
MOV ES, AX
MOV CX, 04H
CLD
REP MOVSB
MOV AX, 3000H
MOV DS, AX
MOV AX, 2000H
MOV ES, AX
MOV SI, 0000H
MOV DI, 0000H
MOV CX, 04H
CLD
REP MOVSB
original source co...
INT 03H
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
2000:0000
F400:0104
BIOS DI
INT 0FFh
IRET
screen source reset aux vars debug stack flags
Random Access Memory
2000:0000 update table list
line: 1 col: 1
drag a file here to open
ENG IN 22:21 26-02-2022


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
;Name - Kondrolla Dinesh Reddy
;Roll no. - 2020BTechCSE040
;Batch - B
;6Q
CODE SEGMENT
ASSUME CS:CODE
MOV AX, 2000H
MOV DS, AX
MOV AX, 3000H
MOV ES, AX
MOV [0000H], 0FH
MOV [0001H], 0EH
MOV [0002H], 0BH
MOV [0003H], 0AH
MOV ES:[0000H], 08H
MOV ES:[0001H], 07H
MOV ES:[0002H], 06H
MOV ES:[0003H], 05H
MOV AX, 2000H
MOV DS, AX
MOV AX, 4000H
MOV SI, 0000H
MOV DI, 0000H
MOV ES, AX
MOV CX, 04H
CLD
REP MOVSB
MOV AX, 3000H
MOV DS, AX
MOV AX, 2000H
MOV ES, AX
MOV SI, 0000H
MOV DI, 0000H
MOV CX, 04H
CLD
REP MOVSB
original source co...
INT 03H
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
3000:0000
F400:0104
BIOS DI
INT 0FFh
IRET
screen source reset aux vars debug stack flags
Random Access Memory
3000:0000 update table list
line: 1 col: 1
drag a file here to open
ENG IN 22:22 26-02-2022

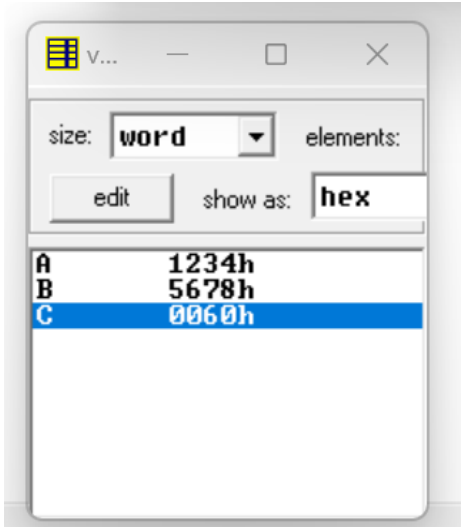
EXPERIMENT – 5:

AIM: Write an ALP to multiply two 16-bit numbers. Operands and result in Data Segment.

CODE:



RESULT:

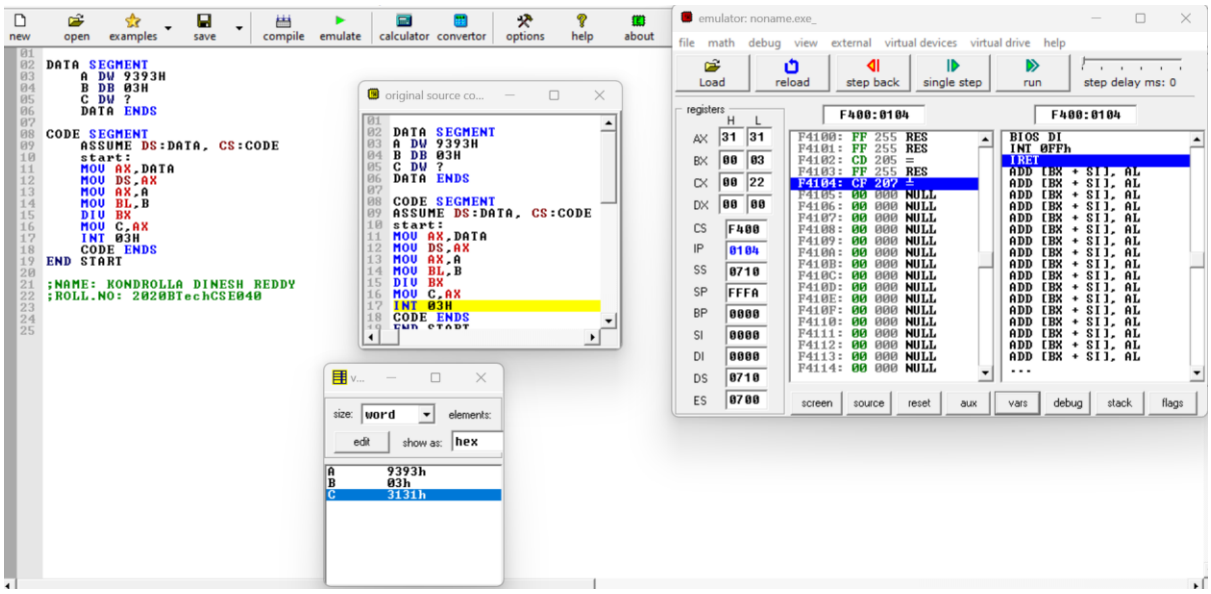


PART-B:

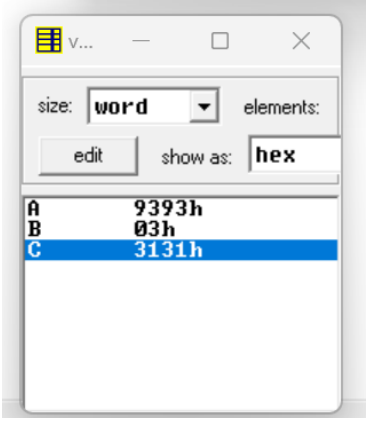
AIM: Write an ALP to divide a 16-bit number by an 8 bit number.

CODE:

```
DATA SEGMENT
     A DW 9393H
     B DB 03H
     C DW ?
     DATA ENDS

CODE SEGMENT
     ASSUME DS:DATA, CS:CODE
     start:
     MOV AX,DATA
     MOV DS,AX
     MOV AX,A
     MOV BL,B
     DIV BX
     MOV C,AX
     INT 03H
     CODE ENDS
END START

;NAME: KONDROLLA DINESH REDDY
;ROLL.NO: 2020BTechCSE040
```



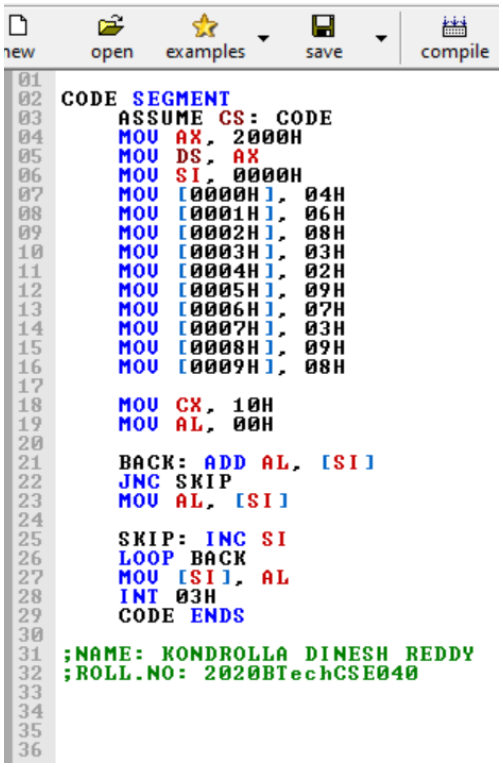
RESULT:



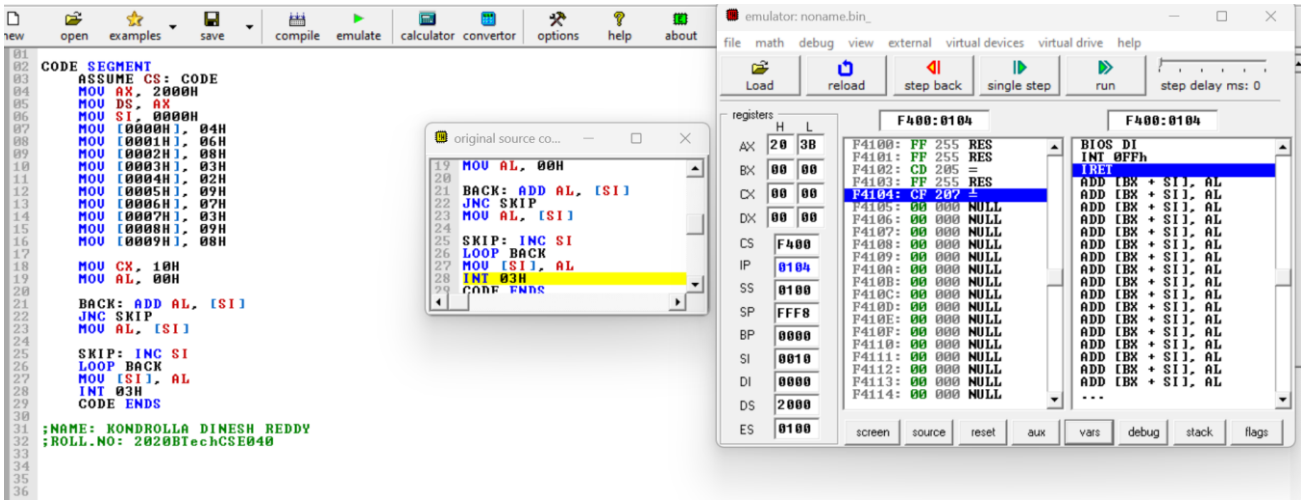
EXPERIMENT – 6:

AIM: Write an ALP to add a series of 10 bytes stored in the memory from locations 20,000H to 20,009H. Store the result immediately after the series.

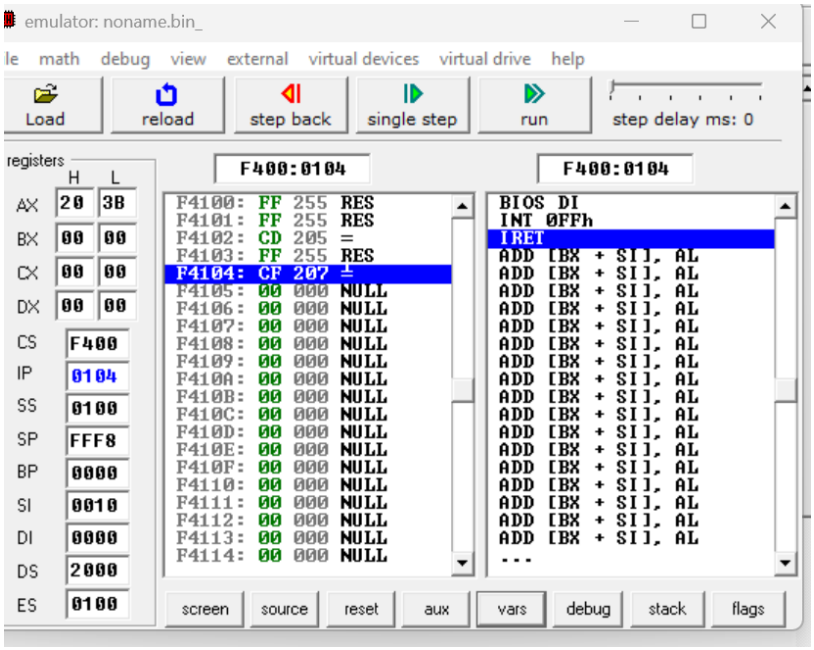
CODE:




```
CODE SEGMENT
    ASSUME CS: CODE
    MOV AX, 2000H
    MOV DS, AX
    MOV SI, 0000H
    MOV [0000H], 04H
    MOV [0001H], 06H
    MOV [0002H], 08H
    MOV [0003H], 03H
    MOV [0004H], 02H
    MOV [0005H], 09H
    MOV [0006H], 07H
    MOV [0007H], 03H
    MOV [0008H], 09H
    MOV [0009H], 08H

    MOV CX, 10H
    MOV AL, 00H

    BACK: ADD AL, [SI]
    JNC SKIP
    MOV AL, [SI]

    SKIP: INC SI
    LOOP BACK
    MOV [SI], AL
    INT 03H
    CODE ENDS

;NAME: KONDROLLA DINESH REDDY
;ROLL.NO: 2020BTechCSE040
```



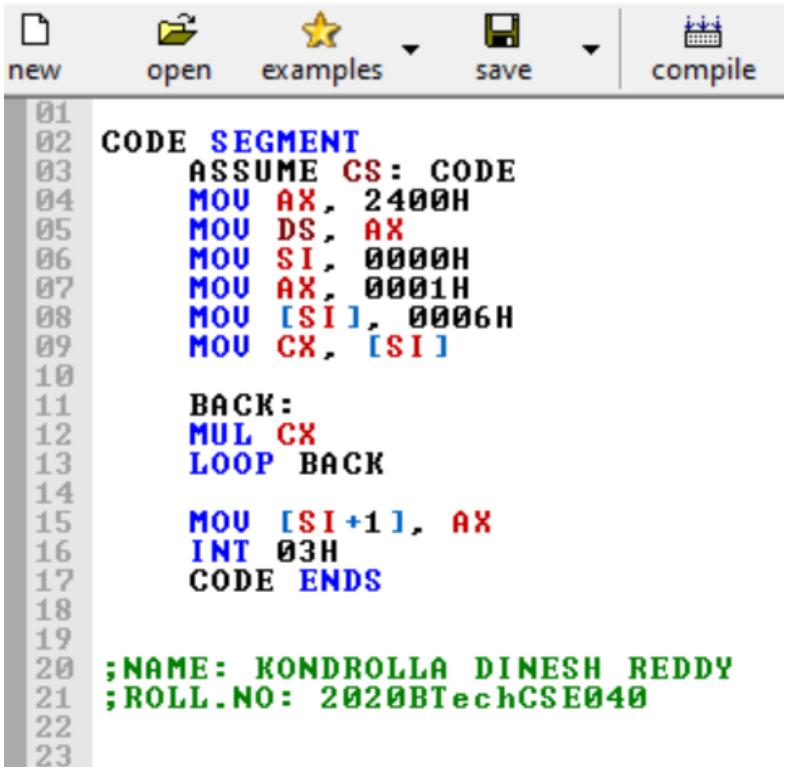
RESULT:

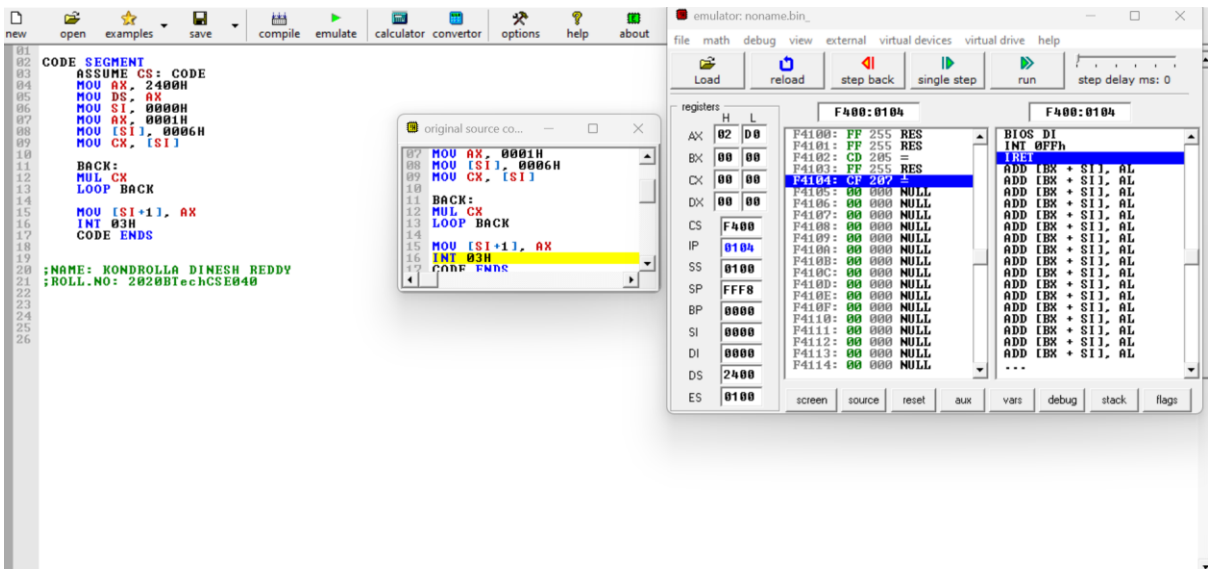

emulator: noname.bin_
le math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 20 3B
BX 00 00
CX 00 00
DX 00 00
CS F400
IP 0104
SS 0100
SP FFF8
BP 0000
SI 0010
DI 0000
DS 2000
ES 0100
F400:0104
F400:0104
F4100: FF 255 RES
F4101: FF 255 RES
F4102: CD 205 =
F4103: FF 255 RES
F4104: CF 207 ╧
F4105: 00 000 NULL
F4106: 00 000 NULL
F4107: 00 000 NULL
F4108: 00 000 NULL
F4109: 00 000 NULL
F410A: 00 000 NULL
F410B: 00 000 NULL
F410C: 00 000 NULL
F410D: 00 000 NULL
F410E: 00 000 NULL
F410F: 00 000 NULL
F4110: 00 000 NULL
F4111: 00 000 NULL
F4112: 00 000 NULL
F4113: 00 000 NULL
F4114: 00 000 NULL
BIOS DI
INT 0FFh
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
---
screen
source
reset
aux
vars
debug
stack
flags

EXPERIMENT – 7:

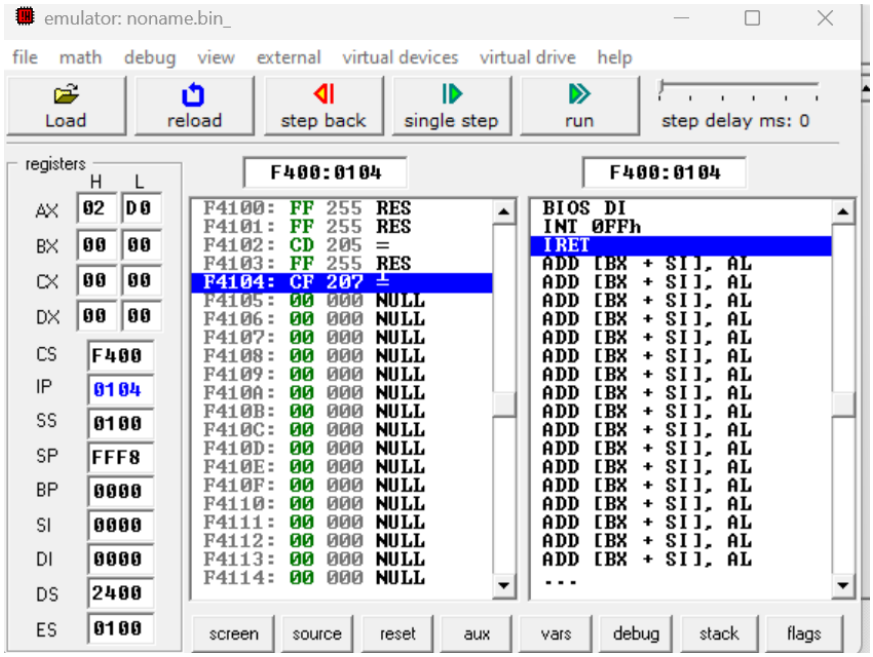
AIM: Write an ALP to find the factorial of a number stored at 24,000H in data segment. Store the result at 24,001H and 24,002H.

CODE:





RESULT:


emulator: noname.bin_
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 02 D0
BX 00 00
CX 00 00
DX 00 00
CS F400
IP 0104
SS 0100
SP FFF8
BP 0000
SI 0000
DI 0000
DS 2400
ES 0100
F400:0104
F4100: FF 255 RES
F4101: FF 255 RES
F4102: CD 205 =
F4103: FF 255 RES
F4104: CF 207 ±
F4105: 00 000 NULL
F4106: 00 000 NULL
F4107: 00 000 NULL
F4108: 00 000 NULL
F4109: 00 000 NULL
F410A: 00 000 NULL
F410B: 00 000 NULL
F410C: 00 000 NULL
F410D: 00 000 NULL
F410E: 00 000 NULL
F410F: 00 000 NULL
F4110: 00 000 NULL
F4111: 00 000 NULL
F4112: 00 000 NULL
F4113: 00 000 NULL
F4114: 00 000 NULL
F400:0104
BIOS DI
INT 0FFh
IRET
ADD [BX + SI], AL
screen
source
reset
aux
vars
debug
stack
flags

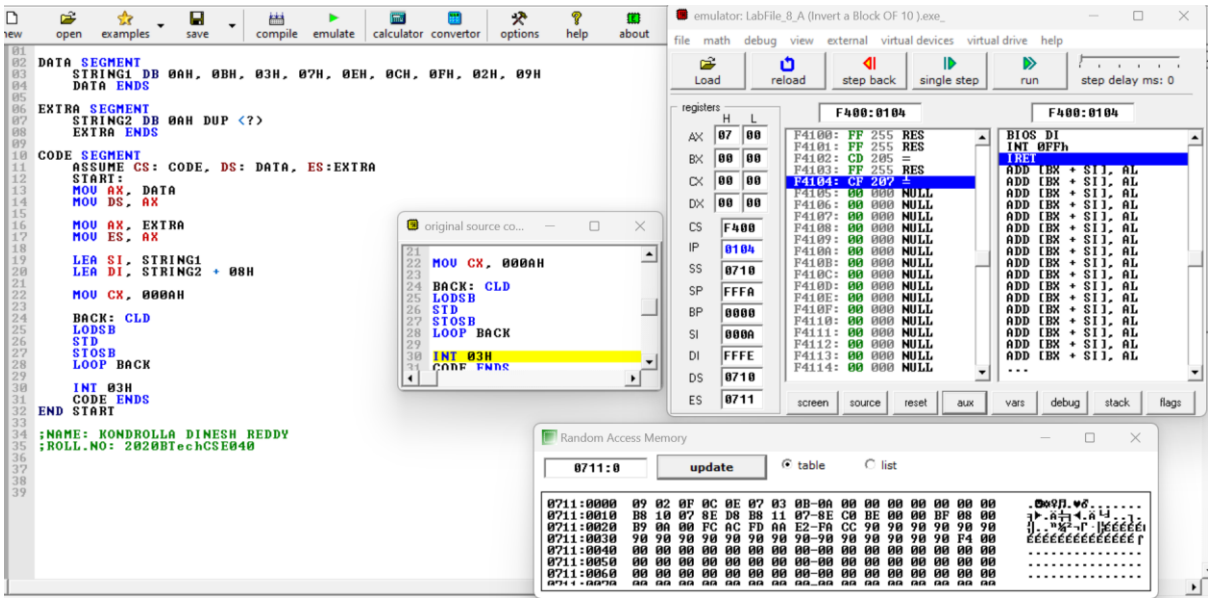
EXPERIMENT-8:

PART-A

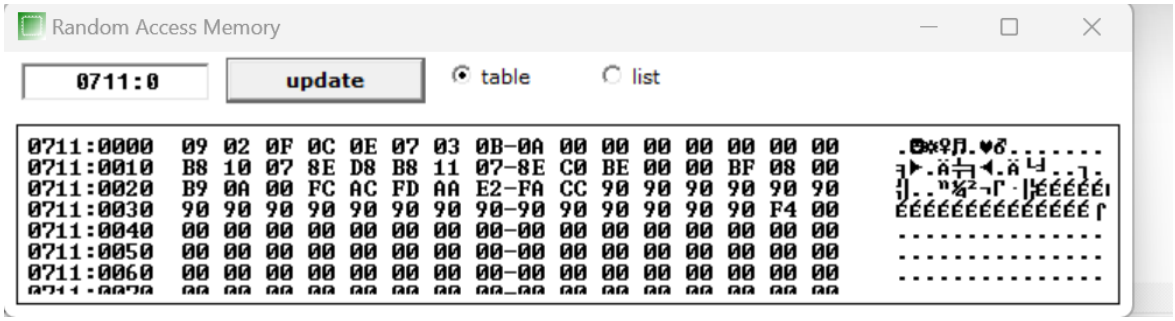
AIM: Write an ALP to invert a block of 10 bytes from Data Segment to Extra Segment**.**

CODE:

```
DATA SEGMENT
    STRING1 DB 0AH, 0BH, 03H, 07H, 0EH, 0CH, 0FH, 02H, 09H
    DATA ENDS

EXTRA SEGMENT
    STRING2 DB 0AH DUP <?>
    EXTRA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DATA, ES:EXTRA
    START:
    MOV AX, DATA
    MOV DS, AX

    MOV AX, EXTRA
    MOV ES, AX

    LEA SI, STRING1
    LEA DI, STRING2 + 08H

    MOV CX, 000AH

    BACK: CLD
    LODSB
    STD
    STOSB
    LOOP BACK

    INT 03H
    CODE ENDS
END START

;NAME: KONDROLLA DINESH REDDY
;ROLL.NO: 2020BTechCSE040
```



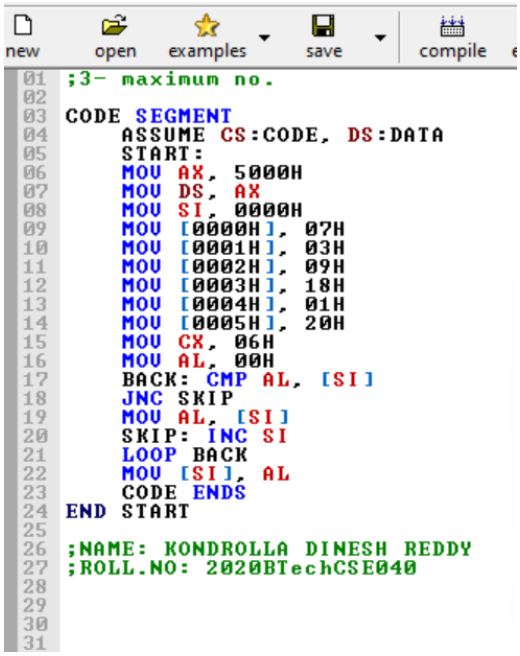
RESULT:

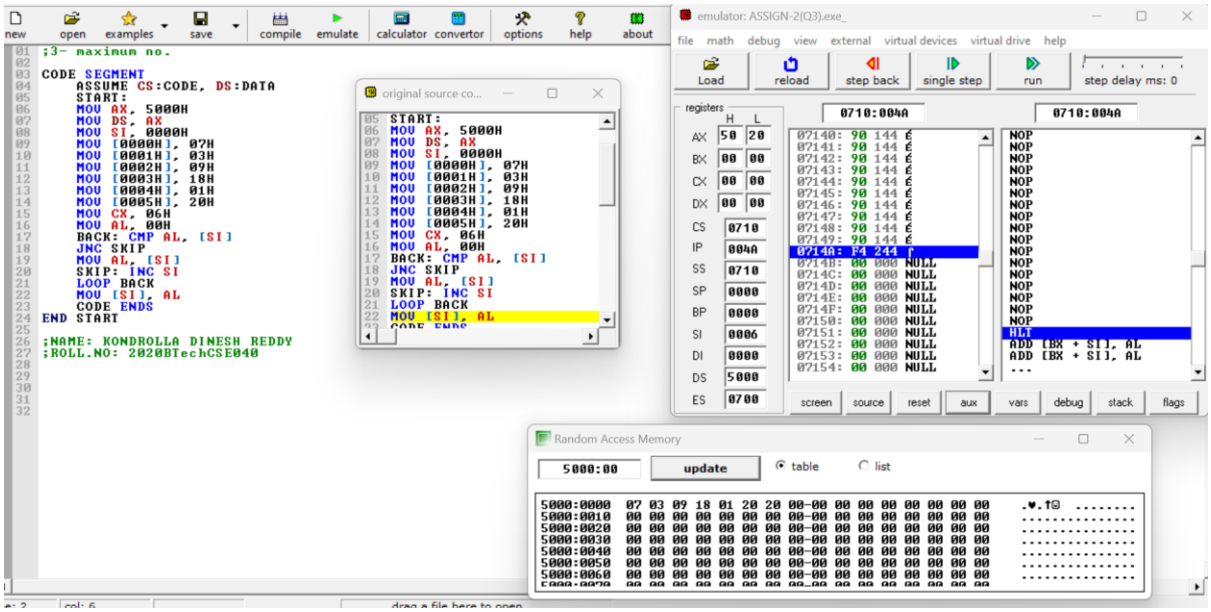


PART-B:

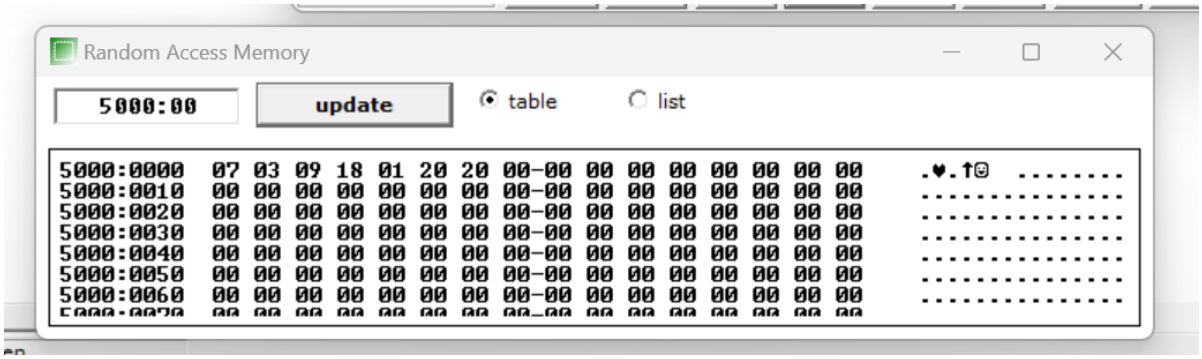
AIM: Write an ALP to get 5 byte of data from the user and store in it memory location 50000H to 50005 H. Find the maximum of it and store it next location.

CODE:




;3- maximun no.
CODE SEGMENT
ASSUME CS:CODE, DS:DATA
START:
MOV AX, 5000H
MOV DS, AX
MOV SI, 0000H
MOV [0000H], 07H
MOV [0001H], 03H
MOV [0002H], 09H
MOV [0003H], 18H
MOV [0004H], 01H
MOV [0005H], 20H
MOV CX, 06H
MOV AL, 00H
BACK: CMP AL, [SI]
JNC SKIP
MOV AL, [SI]
SKIP: INC SI
LOOP BACK
MOV [SI], AL
CODE ENDS
END START
;NAME: KONDROLLA DINESH REDDY
;ROLL.NO: 2020BTechCSE040
Random Access Memory
5000:00
update
table
list

RESULT:

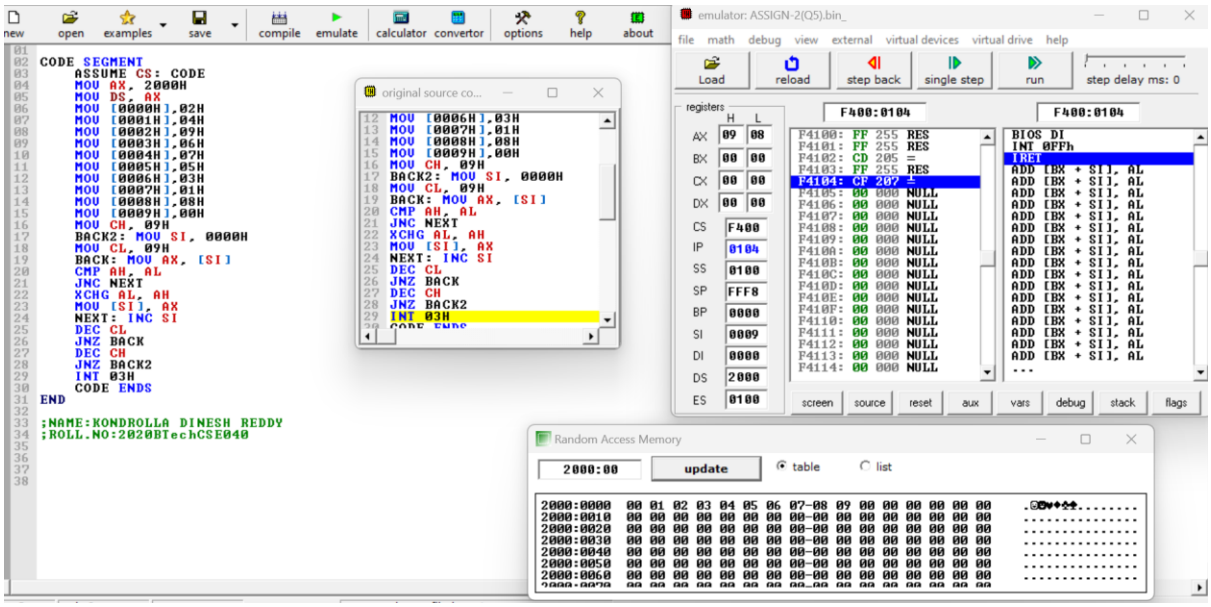

Random Access Memory
5000:00
update
table
list
5000:0000 07 03 09 18 01 20 20 00-00 00 00 00 00 00 00 00
5000:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
5000:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
5000:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
5000:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
5000:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
5000:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

EXPERIMENT – 9:

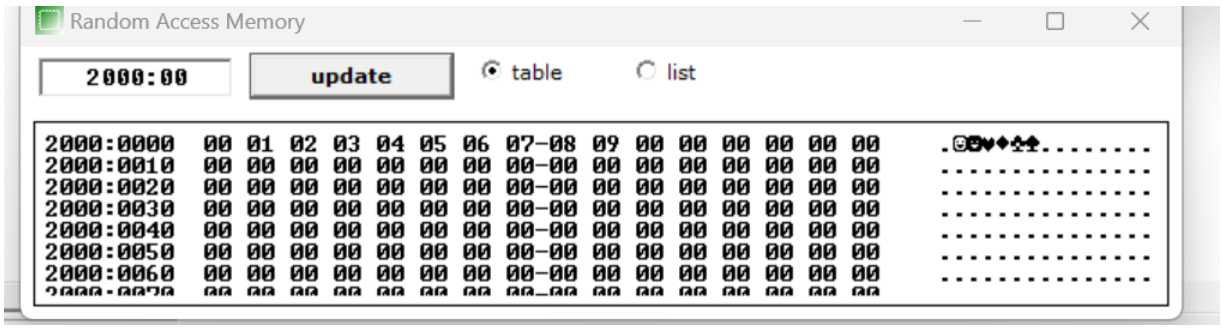
AIM: Write an ALP to SORT a series of 10 numbers from 20,000H in ascending order.

CODE:

```
CODE SEGMENT
    ASSUME CS: CODE
    MOV AX, 2000H
    MOV DS, AX
    MOV [0000H],02H
    MOV [0001H],04H
    MOV [0002H],09H
    MOV [0003H],06H
    MOV [0004H],07H
    MOV [0005H],05H
    MOV [0006H],03H
    MOV [0007H],01H
    MOV [0008H],08H
    MOV [0009H],00H
    MOV CH, 09H
    BACK2: MOV SI, 0000H
    MOV CL, 09H
    BACK: MOV AX, [SI]
    CMP AH, AL
    JNC NEXT
    XCHG AL, AH
    MOV [SI], AX
    NEXT: INC SI
    DEC CL
    JNZ BACK
    DEC CH
    JNZ BACK2
    INT 03H
    CODE ENDS
END

;NAME:KONDROLLA DINESH REDDY
;ROLL.NO:2020BTechCSE040
```



RESULT:



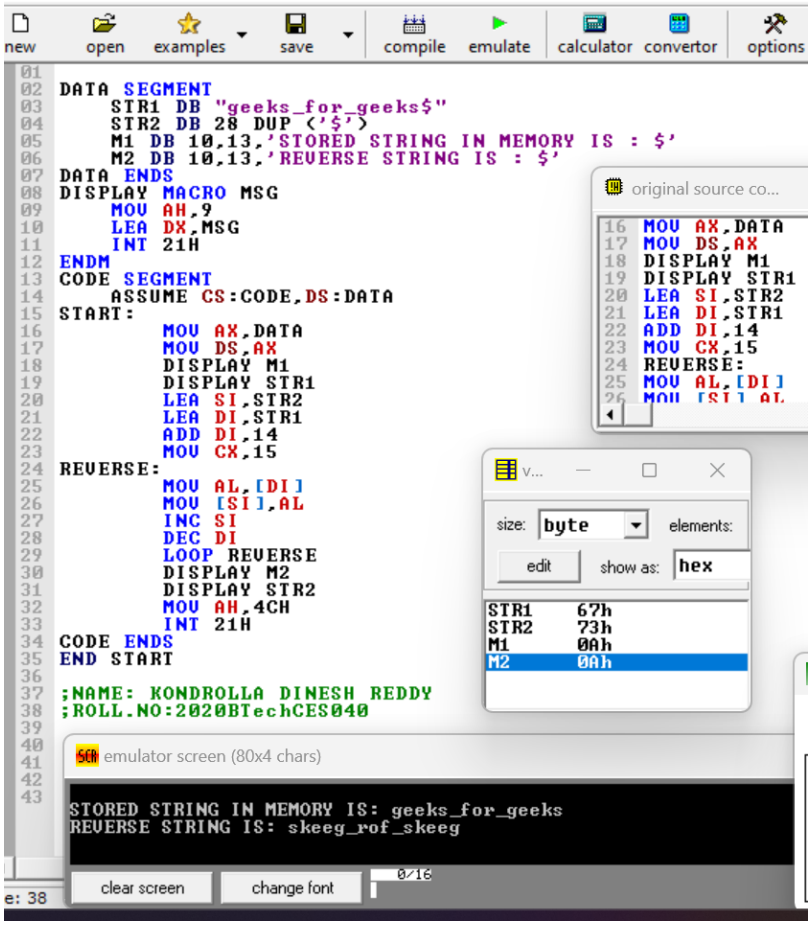
EXPERIMENT – 10:

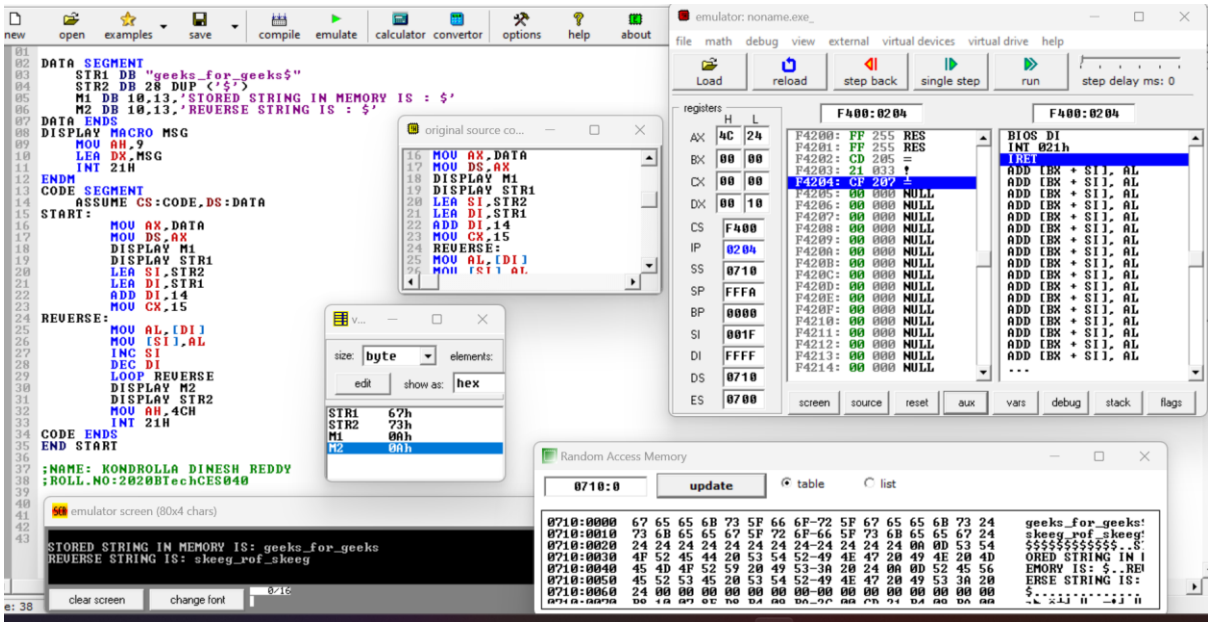
AIM: Write an ALP to reverse the string and print the reversed string.

Ex. Input: String: "Geeks for Geeks"

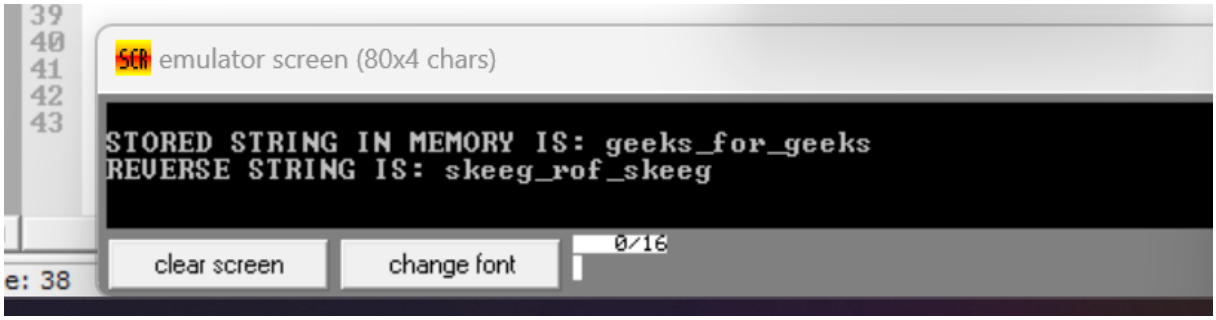
Output: skeeG rof skeeG

CODE:





RESULT:

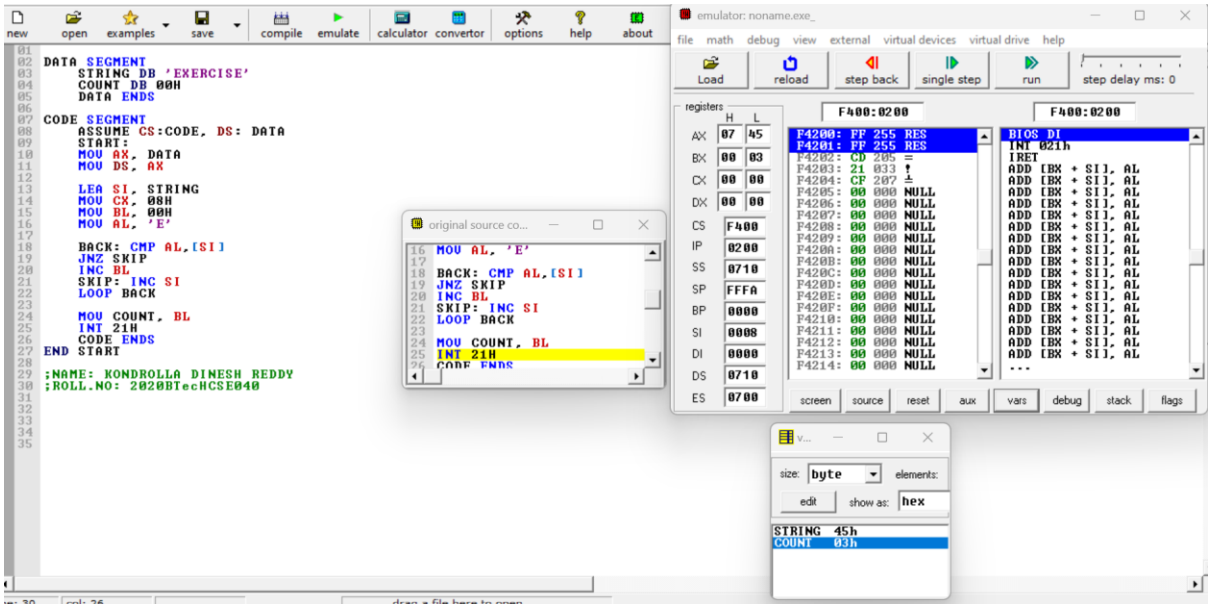

emulator screen (80x4 chars)
STORED STRING IN MEMORY IS: geeks_for_geeks
REVERSE STRING IS: skeeg_rof_skeeg
clear screen
change font
0/16

EXPERIMENT – 11:

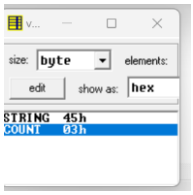
AIM: Write an ALP to determine how many times “e” exists in “exercise”?

CODE:

```
DATA SEGMENT
    STRING DB 'EXERCISE'
    COUNT DB 00H
    DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS: DATA
    START:
    MOV AX, DATA
    MOV DS, AX

    LEA SI, STRING
    MOV CX, 08H
    MOV BL, 00H
    MOV AL, 'E'

    BACK: CMP AL,[SI]
    JNZ SKIP
    INC BL
    SKIP: INC SI
    LOOP BACK

    MOV COUNT, BL
    INT 21H
    CODE ENDS
END START

;NAME: KONDROLLA DINESH REDDY
;ROLL.NO: 2020BTecHCSE040
```



RESULT:



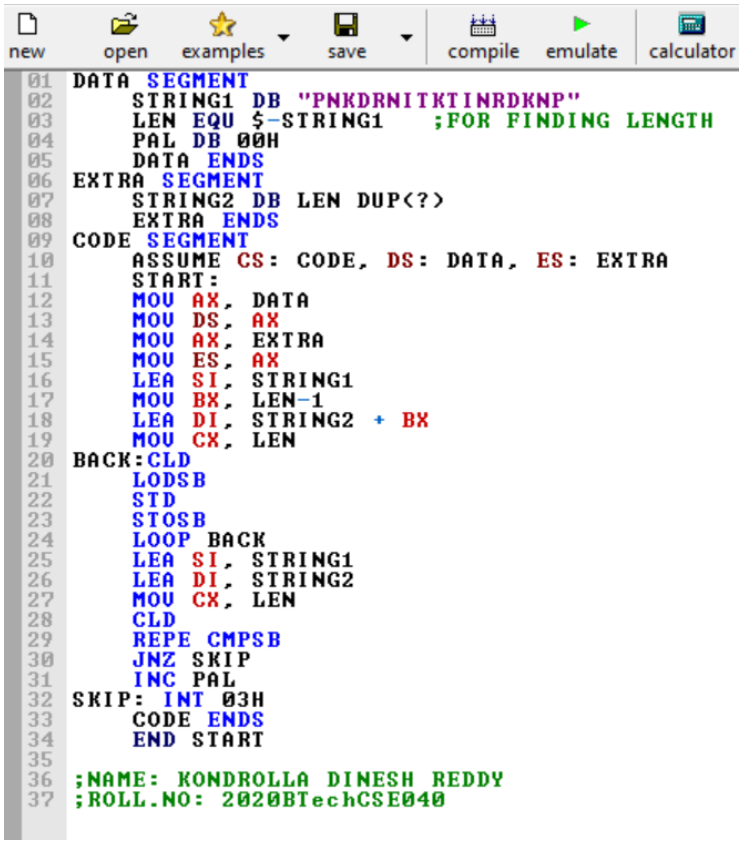
EXPERIMENT – 12:

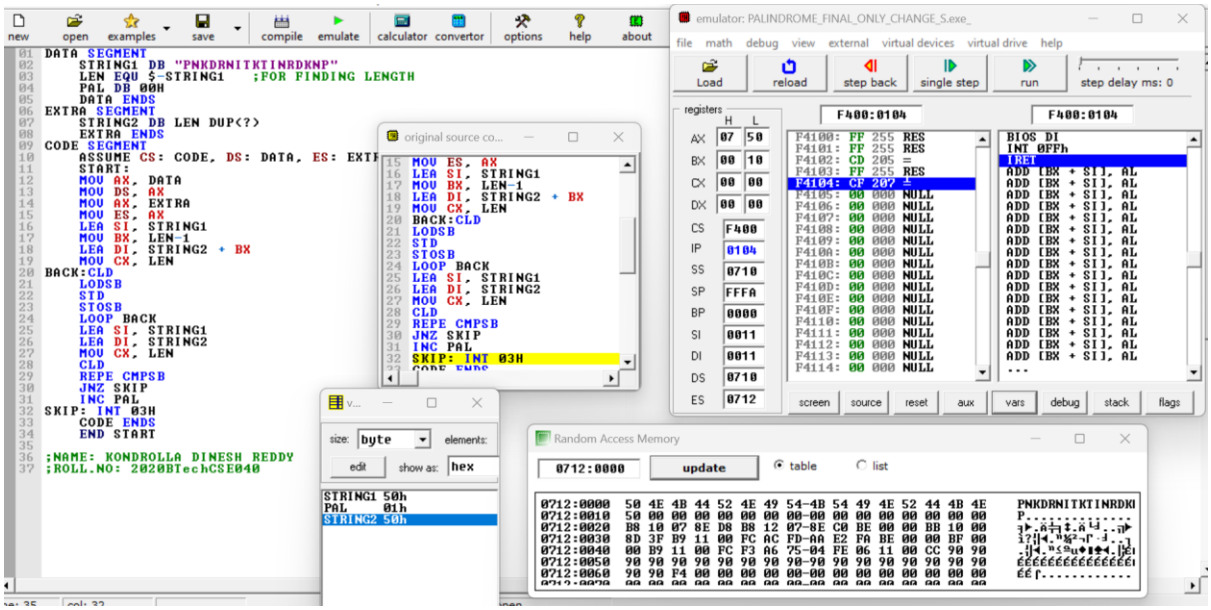
AIM: Write an ALP to given string is palindrome or not?

CODE:

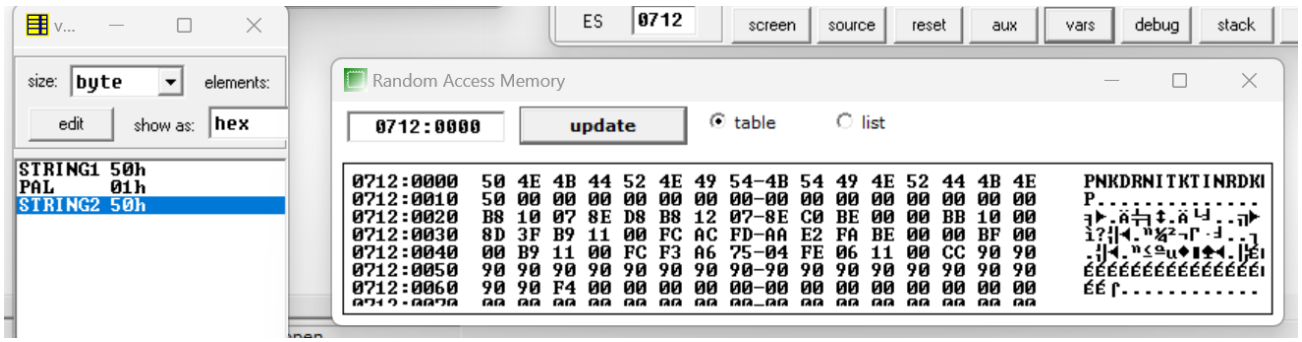
```
DATA SEGMENT
   STRING1 DB "PNKDRNITKTINRDKNP"
   LEN EQU $-STRING1   ;FOR FINDING LENGTH
   PAL DB 00H
   DATA ENDS
EXTRA SEGMENT
   STRING2 DB LEN DUP(?)
   EXTRA ENDS
CODE SEGMENT
   ASSUME CS: CODE, DS: DATA, ES: EXTRA
   START:
   MOV AX, DATA
   MOV DS, AX
   MOV AX, EXTRA
   MOV ES, AX
   LEA SI, STRING1
   MOV BX, LEN-1
   LEA DI, STRING2 + BX
   MOV CX, LEN
BACK:CLD
   LODSB
   STD
   STOSB
   LOOP BACK
   LEA SI, STRING1
   LEA DI, STRING2
   MOV CX, LEN
   CLD
   REPE CMPSB
   JNZ SKIP
```

```
    INC PAL
SKIP: INT 03H
    CODE ENDS
    END START

;NAME: KONDROLLA DINESH REDDY
;ROLL.NO: 2020BTechCSE040
```




DATA SEGMENT
STRING1 DB "PNKDRNITKTINRDKNP"
LEN EQU $-STRING1 ;FOR FINDING LENGTH
PAL DB 00H
DATA ENDS
EXTRA SEGMENT
STRING2 DB LEN DUP(?)
EXTRA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA, ES: EXTR
START:
MOV AX, DATA
MOV DS, AX
MOV AX, EXTRA
MOV ES, AX
LEA SI, STRING1
MOV BX, LEN-1
LEA DI, STRING2 + BX
MOV CX, LEN
BACK:CLD
LODSB
STD
STOSB
LOOP BACK
LEA SI, STRING1
LEA DI, STRING2
MOV CX, LEN
CLD
REPE CMPSB
JNZ SKIP
INC PAL
SKIP: INT 03H
CODE ENDS
END START
;NAME: KONDROLLA DINESH REDDY
;ROLL.NO: 20208TechCSE040
emulator: PALINDROME_FINAL_ONLY_CHANGE_S.exe_
Random Access Memory
0712:0000
update
table
list

RESULT:


ES 0712
screen
source
reset
aux
vars
debug
stack
size: byte
elements:
edit
show as: hex
STRING1 50h
PAL 01h
STRING2 50h
Random Access Memory
0712:0000
update
table
list
0712:0000 50 4E 4B 44 52 4E 49 54-4B 54 49 4E 52 44 4B 4E PNKDRNITKTINRDKI
0712:0010 50 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 P...............
0712:0020 B8 10 07 8E D8 B8 12 07-8E C0 BE 00 00 BB 10 00
0712:0030 8D 3F B9 11 00 FC AC FD-AA E2 FA BE 00 00 BF 00
0712:0040 00 B9 11 00 FC F3 A6 75-04 FE 06 11 00 CC 90 90
0712:0050 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 90
0712:0060 90 90 F4 00 00 00 00 00-00 00 00 00 00 00 00 00

EXPERIMENT – 13:

AIM: Write an ALP to Convert a Decimal Number into Hexadecimal. Assume the Decimal Number is stored at 24000H. Store the result at 24001H.

CODE: