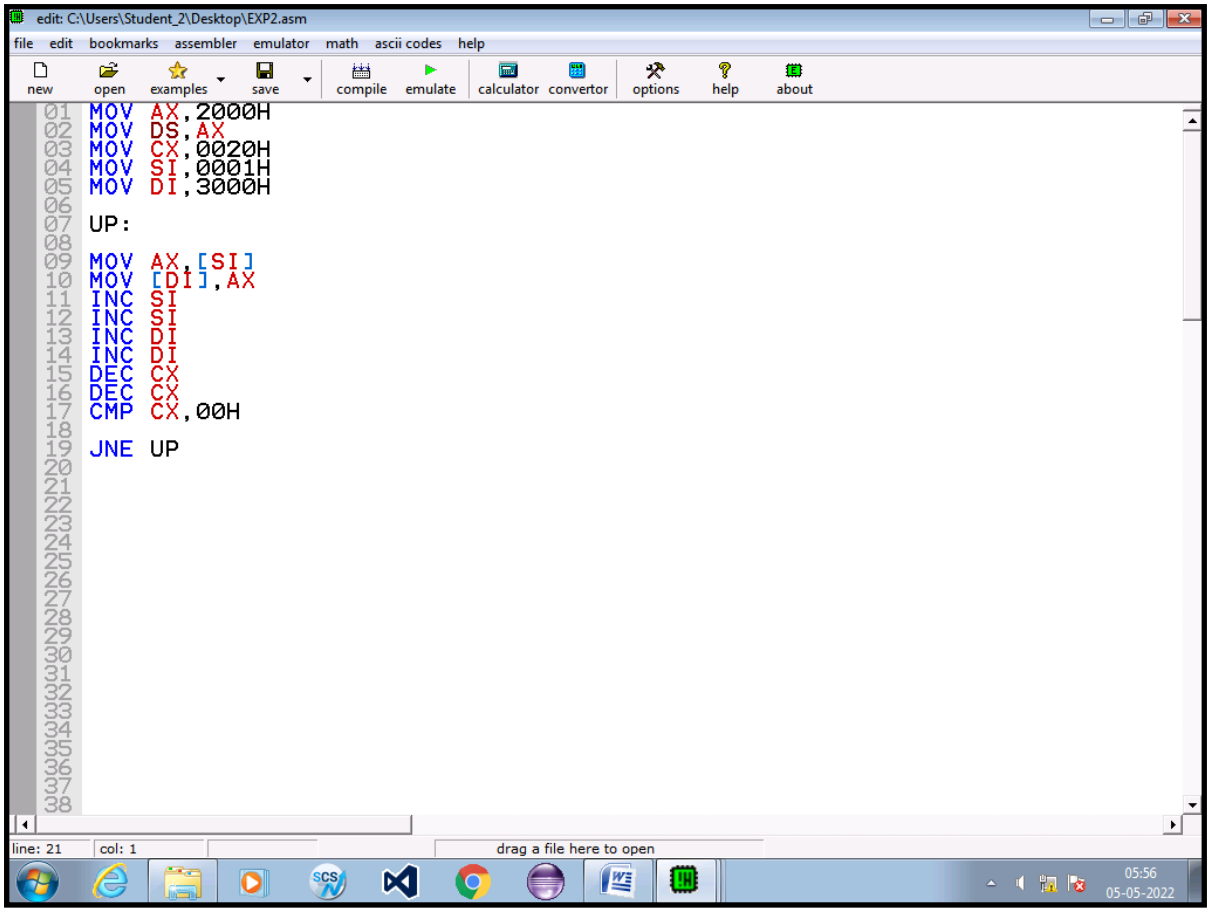
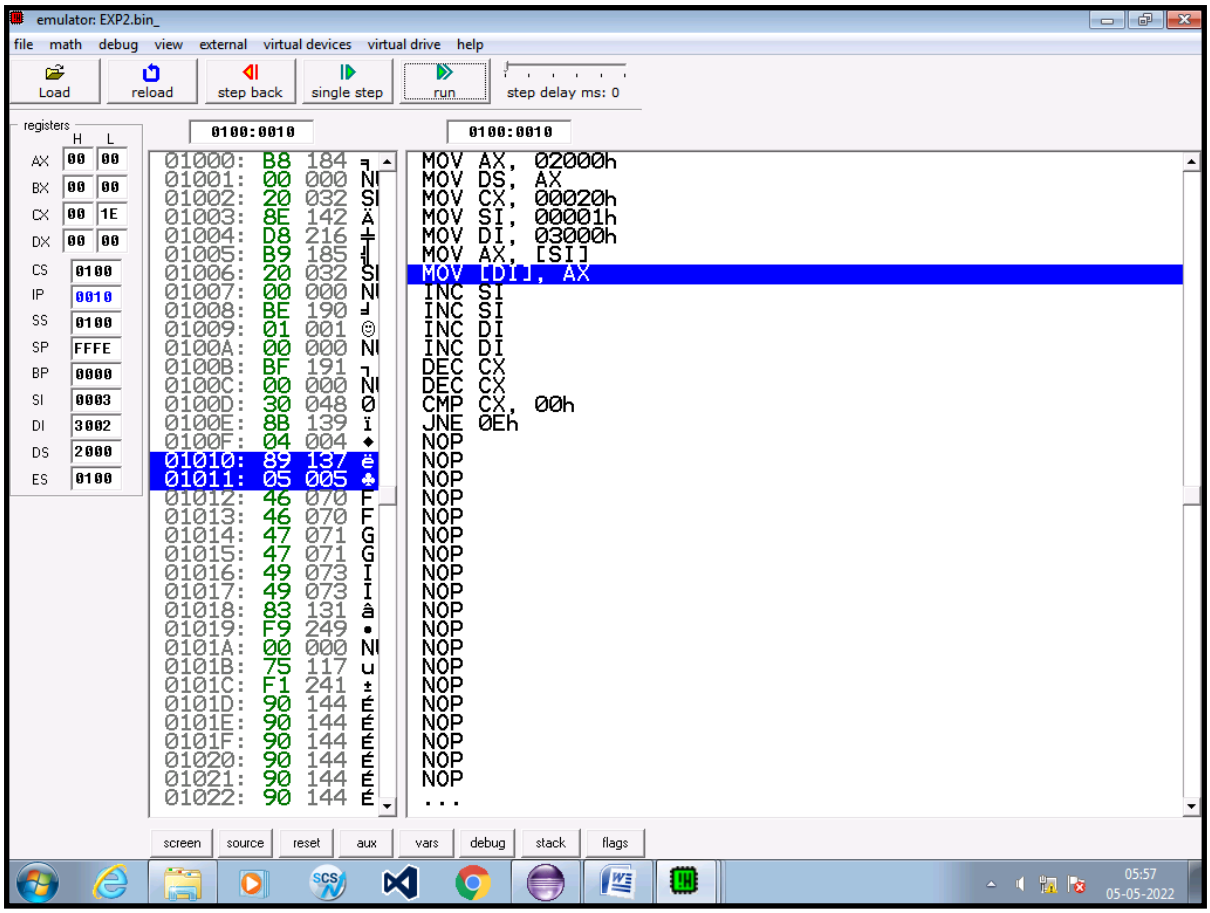
Experiment -2

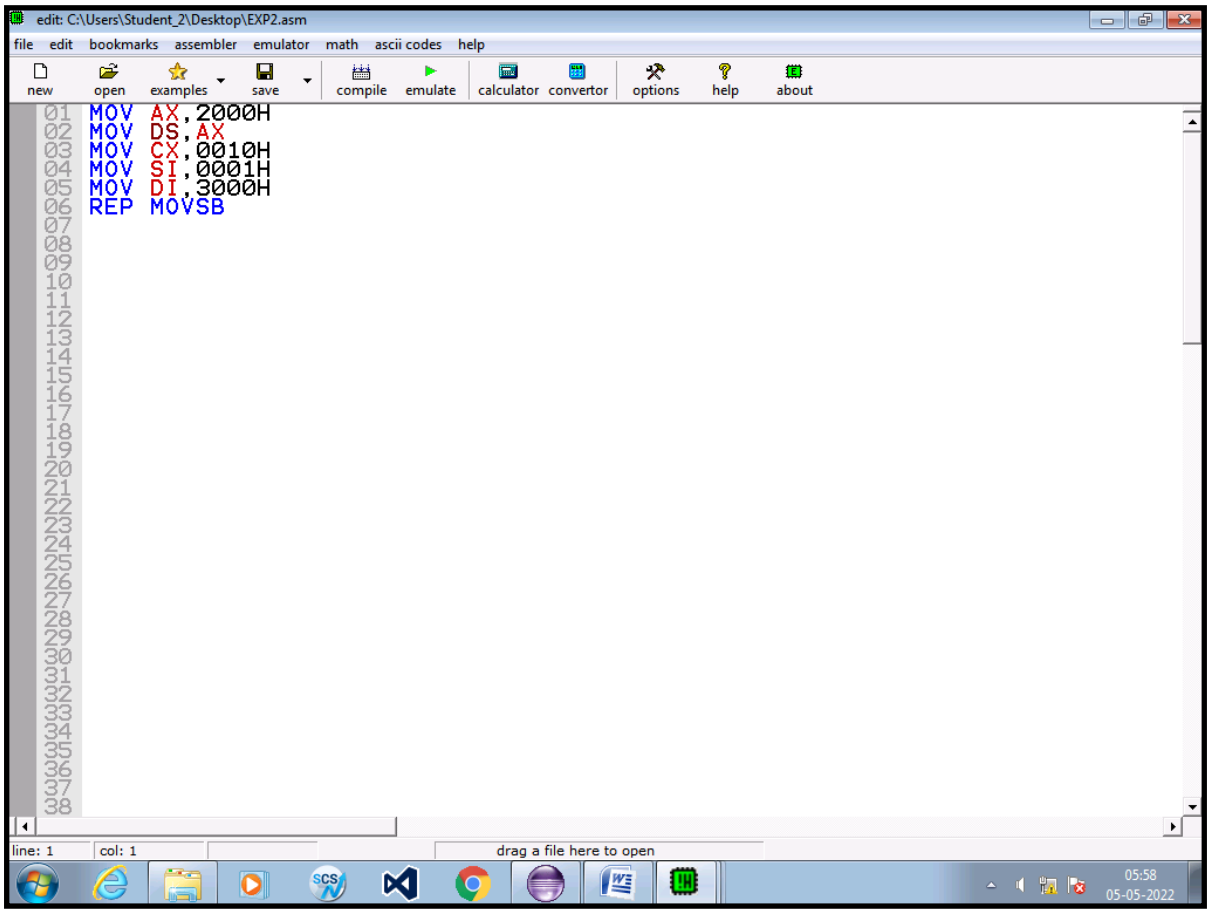
Write an assembly language program in 8086 to transfer the contents of series of numbers stored from 2000:0000H. Length of series is stored at CX register.

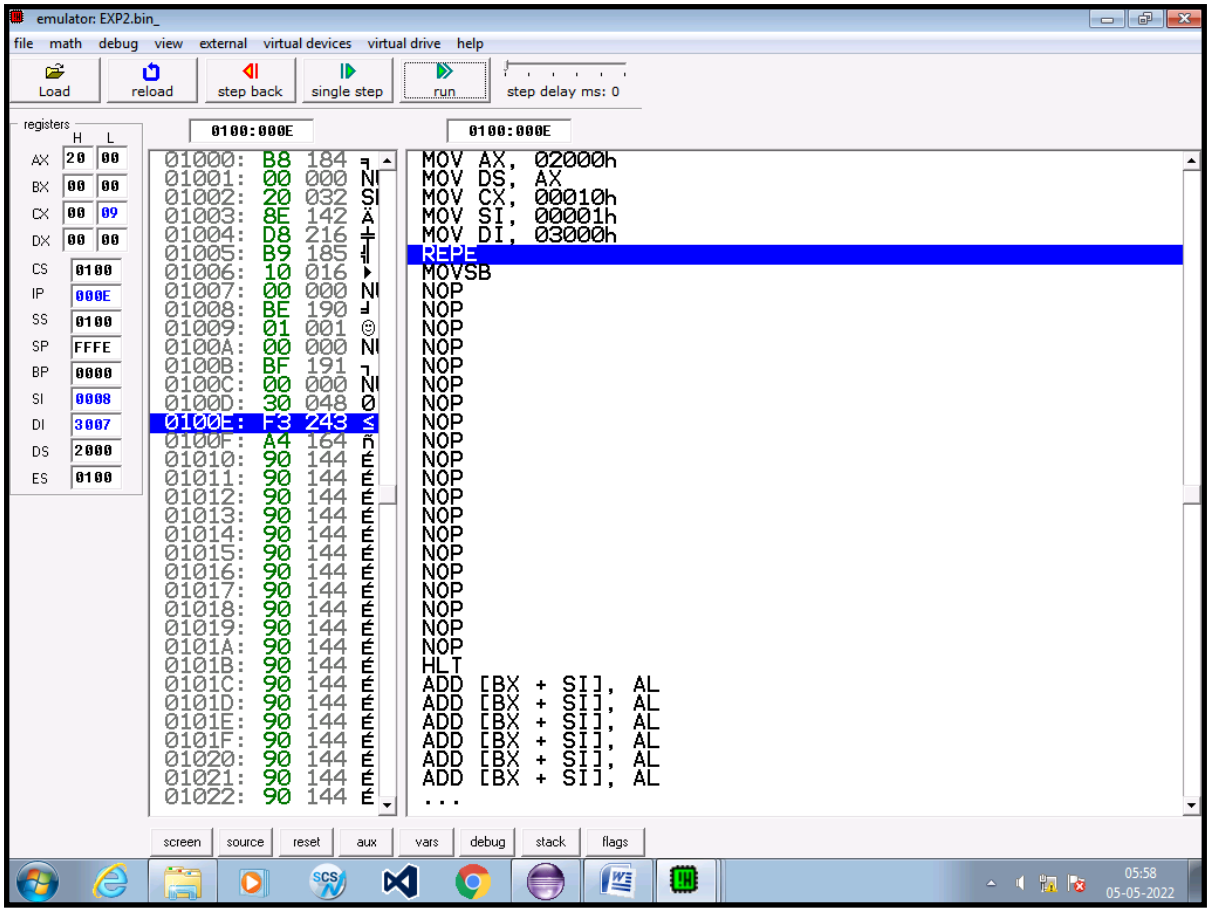
A) Using MOV instruction and looping using JMP


edit: C:\Users\Student_2\Desktop\EXP2.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 MOV AX,2000H
02 MOV DS,AX
03 MOV CX,0020H
04 MOV SI,0001H
05 MOV DI,3000H
06
07 UP:
08
09 MOV AX,[SI]
10 MOV [DI],AX
11 INC SI
12 INC SI
13 INC DI
14 INC DI
15 DEC CX
16 DEC CX
17 CMP CX,00H
18
19 JNE UP
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
line: 21
col: 1
drag a file here to open
05:56
05-05-2022


emulator: EXP2.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 00 00
BX 00 00
CX 00 1E
DX 00 00
CS 0100
IP 0010
SS 0100
SP FFFE
BP 0000
SI 0003
DI 3002
DS 2000
ES 0100
0100:0010
0100:0010
01000: B8 184 ¸
01001: 00 000 N
01002: 20 032 S
01003: 8E 142 Ä
01004: D8 216 ╪
01005: B9 185 ╣
01006: 20 032 S
01007: 00 000 N
01008: BE 190 ╛
01009: 01 001 ☺
0100A: 00 000 N
0100B: BF 191 ┐
0100C: 00 000 N
0100D: 30 048 0
0100E: 8B 139 ï
0100F: 04 004 ♦
01010: 89 137 ë
01011: 05 005 ♣
01012: 46 070 F
01013: 46 070 F
01014: 47 071 G
01015: 47 071 G
01016: 49 073 I
01017: 49 073 I
01018: 83 131 â
01019: F9 249 •
0101A: 00 000 N
0101B: 75 117 u
0101C: F1 241 ±
0101D: 90 144 É
0101E: 90 144 É
0101F: 90 144 É
01020: 90 144 É
01021: 90 144 É
01022: 90 144 É
MOV AX, 02000h
MOV DS, AX
MOV CX, 00020h
MOV SI, 00001h
MOV DI, 03000h
MOV AX, [SI]
MOV [DI], AX
INC SI
INC SI
INC DI
INC DI
DEC CX
DEC CX
CMP CX, 00h
JNE 0Eh
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen source reset aux vars debug stack flags
05:57
05-05-2022

B) Using string instruction and REP prefix




emulator: EXP2.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 20 00
BX 00 00
CX 00 09
DX 00 00
CS 0100
IP 000E
SS 0100
SP FFFE
BP 0000
SI 0008
DI 3007
DS 2000
ES 0100
0100:000E
0100:000E
01000: B8 184
01001: 00 000
01002: 20 032
01003: 8E 142
01004: D8 216
01005: B9 185
01006: 10 016
01007: 00 000
01008: BE 190
01009: 01 001
0100A: 00 000
0100B: BF 191
0100C: 00 000
0100D: 30 048
0100E: F3 243
0100F: A4 164
01010: 90 144
01011: 90 144
01012: 90 144
01013: 90 144
01014: 90 144
01015: 90 144
01016: 90 144
01017: 90 144
01018: 90 144
01019: 90 144
0101A: 90 144
0101B: 90 144
0101C: 90 144
0101D: 90 144
0101E: 90 144
0101F: 90 144
01020: 90 144
01021: 90 144
01022: 90 144
MOV AX, 02000h
MOV DS, AX
MOV CX, 00010h
MOV SI, 00001h
MOV DI, 03000h
REPE
MOVSB
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
HLT
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
05:58
05-05-2022

C) Repeat a and b above when the numbers are stored from 2000:0001H