

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

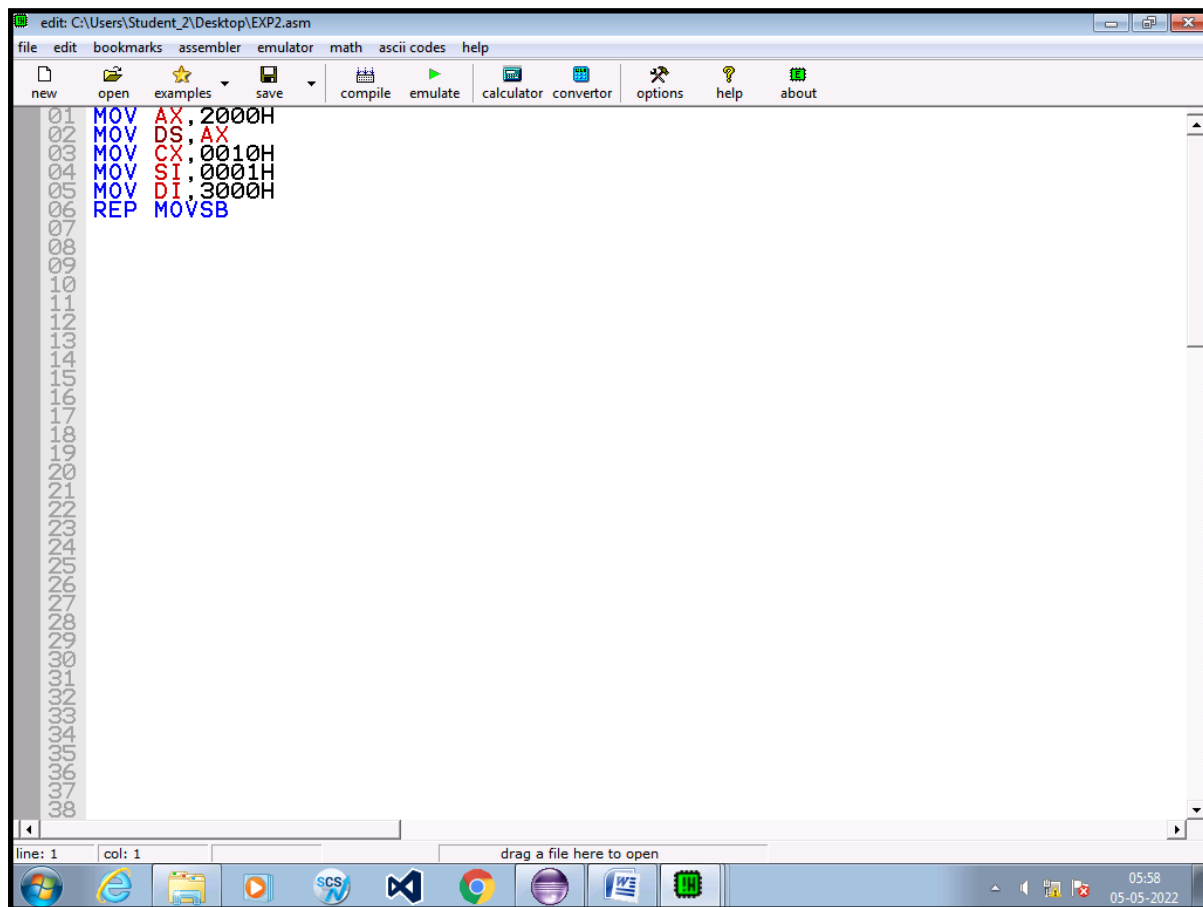
Load reload step back single step run step delay ms: 0

registers	H	L	0100:0010	0100:0010
AX	00	00	01000: B8 184	MOV AX, 02000h
BX	00	00	01001: 00 000	MOV DS, AX
CX	00	1E	01002: 20 032	MOV CX, 00020h
DX	00	00	01003: 8E 142	MOV SI, 00001h
CS	0100		01004: D8 216	MOV DI, 03000h
IP	0010		01005: B9 185	MOV AX, [SI]
SS	0100		01006: 20 032	MOV [DI], AX
SP	FFFE		01007: 00 000	INC SI
BP	0000		01008: BE 190	INC SI
SI	0003		01009: 01 001	INC DI
DI	3002		0100A: 00 000	INC DI
DS	2000		0100B: BF 191	DEC CX
ES	0100		0100C: 00 000	DEC CX
			0100D: 30 048	CMP CX, 00h
			0100E: 8B 139	JNE 0Eh
			0100F: 04 004	NOP
			01010: 89 137	NOP
			01011: 05 005	NOP
			01012: 46 070	NOP
			01013: 46 070	NOP
			01014: 47 071	NOP
			01015: 47 071	NOP
			01016: 49 073	NOP
			01017: 49 073	NOP
			01018: 83 131	NOP
			01019: F9 249	NOP
			0101A: 00 000	NOP
			0101B: 75 117	NOP
			0101C: F1 241	NOP
			0101D: 90 144	NOP
			0101E: 90 144	NOP
			0101F: 90 144	NOP
			01020: 90 144	NOP
			01021: 90 144	NOP
			01022: 90 144	NOP

screen source reset aux vars debug stack flags

05:57 05-05-2022

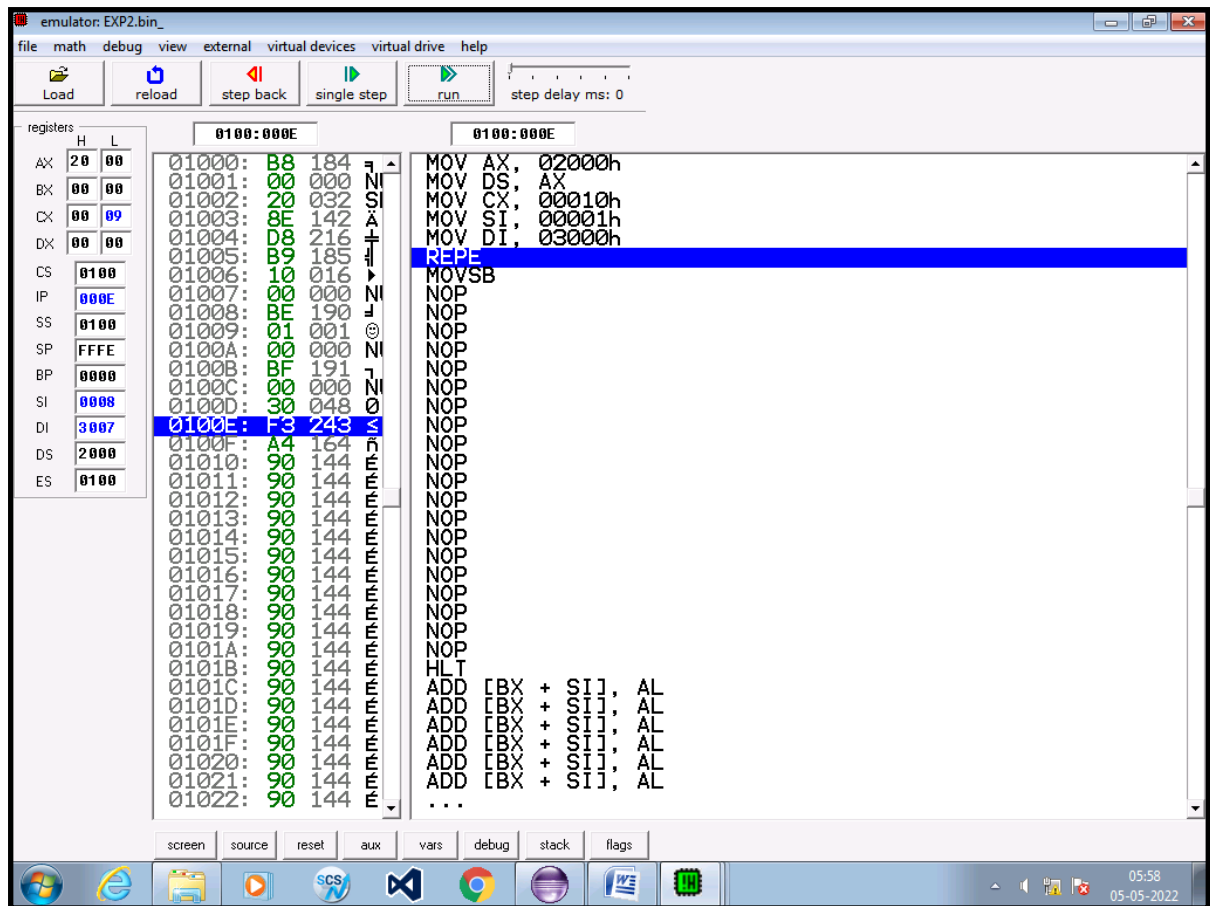
B) Using string instruction and REP prefix



The screenshot shows a window titled "edit: C:\Users\Student_2\Desktop\EXP2.asm". The menu bar includes "file", "edit", "bookmarks", "assembler", "emulator", "math", "ascii codes", and "help". The toolbar contains icons for "new", "open", "examples", "save", "compile", "emulate", "calculator", "converter", "options", "help", and "about". The main text area contains the following assembly code:

```
01 MOV AX,2000H
02 MOV DS,AX
03 MOV CX,0010H
04 MOV SI,0001H
05 MOV DI,3000H
06 REP MOVSB
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

The status bar at the bottom shows "line: 1", "col: 1", and a "drag a file here to open" area. The Windows taskbar at the very bottom includes icons for the Start menu, Internet Explorer, File Explorer, a media player, SCS, Visual Studio Code, Google Chrome, a purple icon, Word, and a green icon. The system clock shows "05:58" and the date "05-05-2022".



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

```

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

001  MOV AX, 2000H
002  MOV DS, AX
003  MOV CX, 0020H
004  MOV SI, 0001H
005  MOV DI, 3000H
006
007  UP:
008
009  MOV AX, [SI]
010  MOV [DI], AX
011  INC SI
012  INC DI
013  INC SI
014  INC DI
015  DEC CX
016  DEC CX
017  CMP CX, 00H
018
019  JNE UP
020
021  MOV AX, 2000H
022  MOV DS, AX
023  MOV CX, 0010H
024  MOV SI, 0001H
025  MOV DI, 3000H
026  REP MOVSB
027
028
029
030
031
032
033
034
035
036
037
038
line: 1 col: 9 drag a file here to open

```

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	00
DX	00	00
CS	0100	
IP	0041	
SS	0100	
SP	FFFE	
BP	0000	
SI	0011	
DI	3010	
DS	2000	
ES	0100	

0100:0041	0100:0041
01025: BE 190	NOP
01026: 01 001	NOP
01027: 00 000	NOP
01028: BF 191	HLL
01029: 00 000	ADD [BX + SI], AL
0102A: 30 048	ADD [BX + SI], AL
0102B: F3 243	ADD [BX + SI], AL
0102C: A4 164	ADD [BX + SI], AL
0102D: 90 144	ADD [BX + SI], AL
0102E: 90 144	ADD [BX + SI], AL
0102F: 90 144	ADD [BX + SI], AL
01030: 90 144	ADD [BX + SI], AL
01031: 90 144	ADD [BX + SI], AL
01032: 90 144	ADD [BX + SI], AL
01033: 90 144	ADD [BX + SI], AL
01034: 90 144	ADD [BX + SI], AL
01035: 90 144	ADD [BX + SI], AL
01036: 90 144	ADD [BX + SI], AL
01037: 90 144	ADD [BX + SI], AL
01038: 90 144	ADD [BX + SI], AL
01039: 90 144	ADD [BX + SI], AL
0103A: 90 144	ADD [BX + SI], AL
0103B: 90 144	ADD [BX + SI], AL
0103C: 90 144	ADD [BX + SI], AL
0103D: 90 144	ADD [BX + SI], AL
0103E: 90 144	ADD [BX + SI], AL
0103F: 90 144	ADD [BX + SI], AL
01040: 90 144	ADD [BX + SI], AL
01041: F4 244	ADD [BX + SI], AL
01042: 00 000	ADD [BX + SI], AL
01043: 00 000	ADD [BX + SI], AL
01044: 00 000	ADD [BX + SI], AL
01045: 00 000	ADD [BX + SI], AL
01046: 00 000	ADD [BX + SI], AL
01047: 00 000	...

screen source reset aux vars debug stack flags

