EMULATOR – SEMESTER PROJECT

Write a computer program to emulate the SC computer. This program, called either a simulator or emulator, is to be written in any programming language of your choice. You must demonstrate this program in the Computer Lab before you turn it in. If you do not use software that is available in the Computer Lab, it is your responsibility to make sure that you will be able to demonstrate.

This assignment will be worth 100 points. There will be a penalty assessed for late submission.

The program will demonstrate:

The following basic emulator tasks (required):

- Keyboard input into memory and input cards
- Keyboard input into program counter
- Protection of cell 00 (Input and Execution)
- Updating cell 99
- Updating the CPU
- Clearing memory, input cards and output cards separately
- Clearing all
- Listing all memory components to the screen
- Implementation of a step or trace command, showing contents of the CPU at each step
- Program halt when an empty input card is encountered

The following emulator file tasks (required):

- File save
- File load
- Successful execution of saved and loaded file

A help screen or screens (required), covering:

- The opcodes
- Using the program

Demonstration of successful execution of the four sample programs (required, see next pages)

Requirements for text-based programs (not GUI-based):

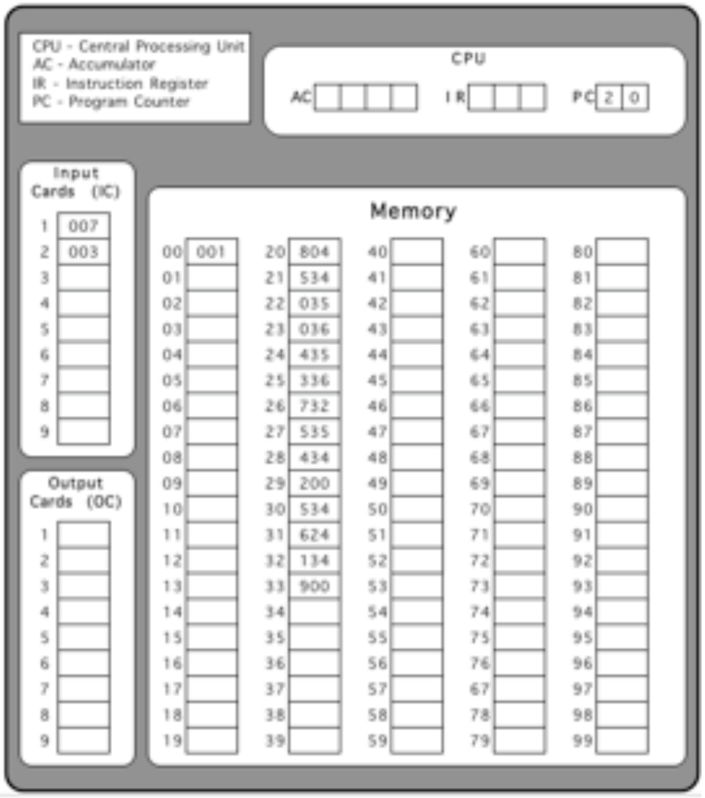
- The user must be able to enter a sequence of instructions into memory after specifying the beginning address (i.e. he/she does not have to specify the address before each instruction)
- Input into memory must be terminated by pressing the ENTER key

The following emulator tasks are optional:

- Adding the name of the emulator, the name of the programmer, the class, the semester and the year to the beginning screen
- Adding a cover sheet and heading pages for each section of the package

The program will demonstrate successful execution of the following programs (required):

Program No. 5: DIVISION (Section 1.3.4, pages 22-23)



CPU - Central Processing Unit
AC - Accumulator
IR - Instruction Register
PC - Program Counter

CPU

AC	IR	PC
		2 0

Input Cards (IC)

1	007
2	003
3	
4	
5	
6	
7	
8	
9	

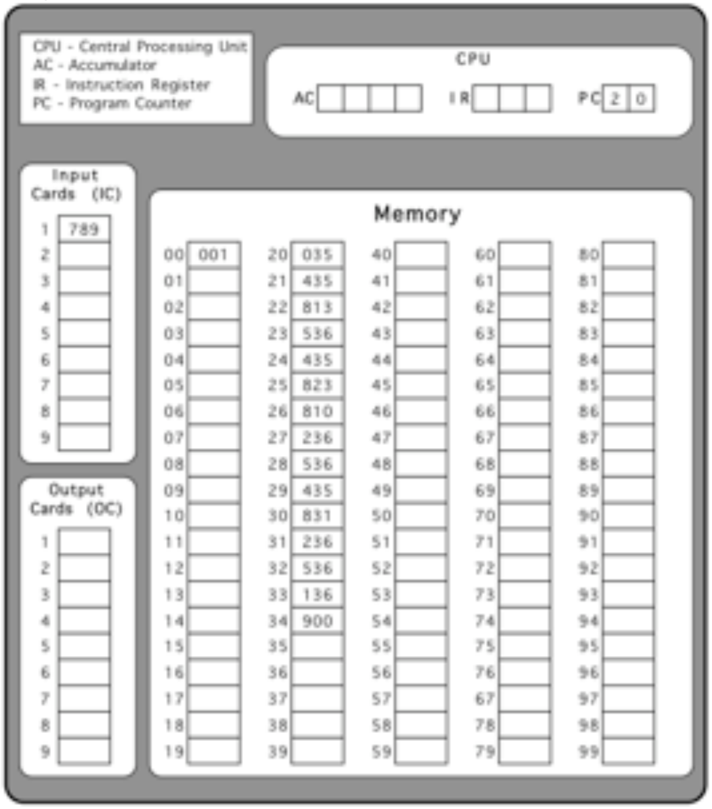
Output Cards (OC)

1	
2	
3	
4	
5	
6	
7	
8	
9	

Memory

00	001	20	804	40		60		80	
01		21	534	41		61		81	
02		22	035	42		62		82	
03		23	036	43		63		83	
04		24	435	44		64		84	
05		25	336	45		65		85	
06		26	732	46		66		86	
07		27	535	47		67		87	
08		28	434	48		68		88	
09		29	200	49		69		89	
10		30	534	50		70		90	
11		31	624	51		71		91	
12		32	134	52		72		92	
13		33	900	53		73		93	
14		34		54		74		94	
15		35		55		75		95	
16		36		56		76		96	
17		37		57		67		97	
18		38		58		78		98	
19		39		59		79		99	

Program No. 6: SHIFTING DIGITS (Section 1.3.5, page 25)



CPU - Central Processing Unit
AC - Accumulator
IR - Instruction Register
PC - Program Counter

CPU

AC	IR	PC
		2 0

Input Cards (IC)

1	789
2	
3	
4	
5	
6	
7	
8	
9	

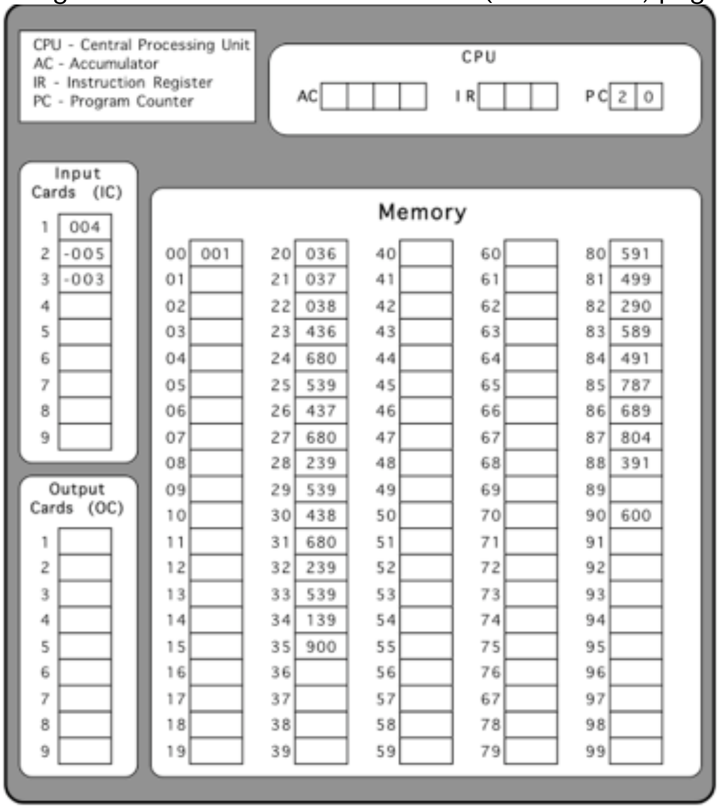
Output Cards (OC)

1	
2	
3	
4	
5	
6	
7	
8	
9	

Memory

00	001	20	035	40		60		80	
01		21	435	41		61		81	
02		22	813	42		62		82	
03		23	536	43		63		83	
04		24	435	44		64		84	
05		25	823	45		65		85	
06		26	810	46		66		86	
07		27	236	47		67		87	
08		28	536	48		68		88	
09		29	435	49		69		89	
10		30	831	50		70		90	
11		31	236	51		71		91	
12		32	536	52		72		92	
13		33	136	53		73		93	
14		34	900	54		74		94	
15		35		55		75		95	
16		36		56		76		96	
17		37		57		67		97	
18		38		58		78		98	
19		39		59		79		99	

Program No. 9 ABSOLUTE VALUE   (Section 1.6, pages 38-39)



CPU - Central Processing Unit
AC - Accumulator
IR - Instruction Register
PC - Program Counter

CPU

AC	IR	PC
		2 0

Input Cards (IC)

1	004
2	-005
3	-003
4	
5	
6	
7	
8	
9	

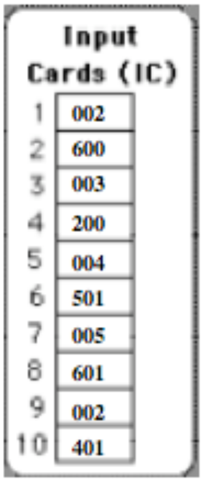
Output Cards (OC)

1	
2	
3	
4	
5	
6	
7	
8	
9	

Memory

00	001	20	036	40		60		80	591
01		21	037	41		61		81	499
02		22	038	42		62		82	290
03		23	436	43		63		83	589
04		24	680	44		64		84	491
05		25	539	45		65		85	787
06		26	437	46		66		86	689
07		27	680	47		67		87	804
08		28	239	48		68		88	391
09		29	539	49		69		89	
10		30	438	50		70		90	600
11		31	680	51		71		91	
12		32	239	52		72		92	
13		33	539	53		73		93	
14		34	139	54		74		94	
15		35	900	55		75		95	
16		36		56		76		96	
17		37		57		67		97	
18		38		58		78		98	
19		39		59		79		99	

Bootstrap the LOADER program from the following input cards:
(nothing in memory initially, and the program counter set to zero)



Input Cards (IC)

1	002
2	600
3	003
4	200
5	004
6	501
7	005
8	601
9	002
10	401

The loader program:

```
02:   401
03:   200
04:   501
05:   601
```

(after execution the loader program will be in cells 2 through 5)

EMULATOR PROJECT HEADER SHEET

Name: ____

Semester: ____________________________

Software/Language used: ____________________________________

Grading:

Functionality of program: (50%) ______________

Ease of use/look of program: (25%) ______________

The project components (a neat, complete package in a binder, organized as follows:) (25%)

____

This project header sheet

A user manual (at least 5 pages) that includes

Typed information that describes clearly to any user how to use the emulator, including what the program does, what type of computer it will execute on, how to start execution, etc.

A print copy of the gui or the main menu and any submenus.

A print copy of the help screen(s)

Specified source code (in class)

Efficiently organized and adequately documented (i.e. easy to read)

Printed demonstration of the four test cases (memory, input cards and output cards before and after execution).

Program No. 5: DIVISION (Section 1.3.4, pages 22-23)

Program No. 6: SHIFTING DIGITS (Section 1.3.5, page 25)

Program No. 9 ABSOLUTE VALUE (Section 1.6, pages 38-39)

Bootstrap the LOADER program from the input cards (from class)

Emulator Progress Worksheets

Extra Credit ______________

GRADE: ______________