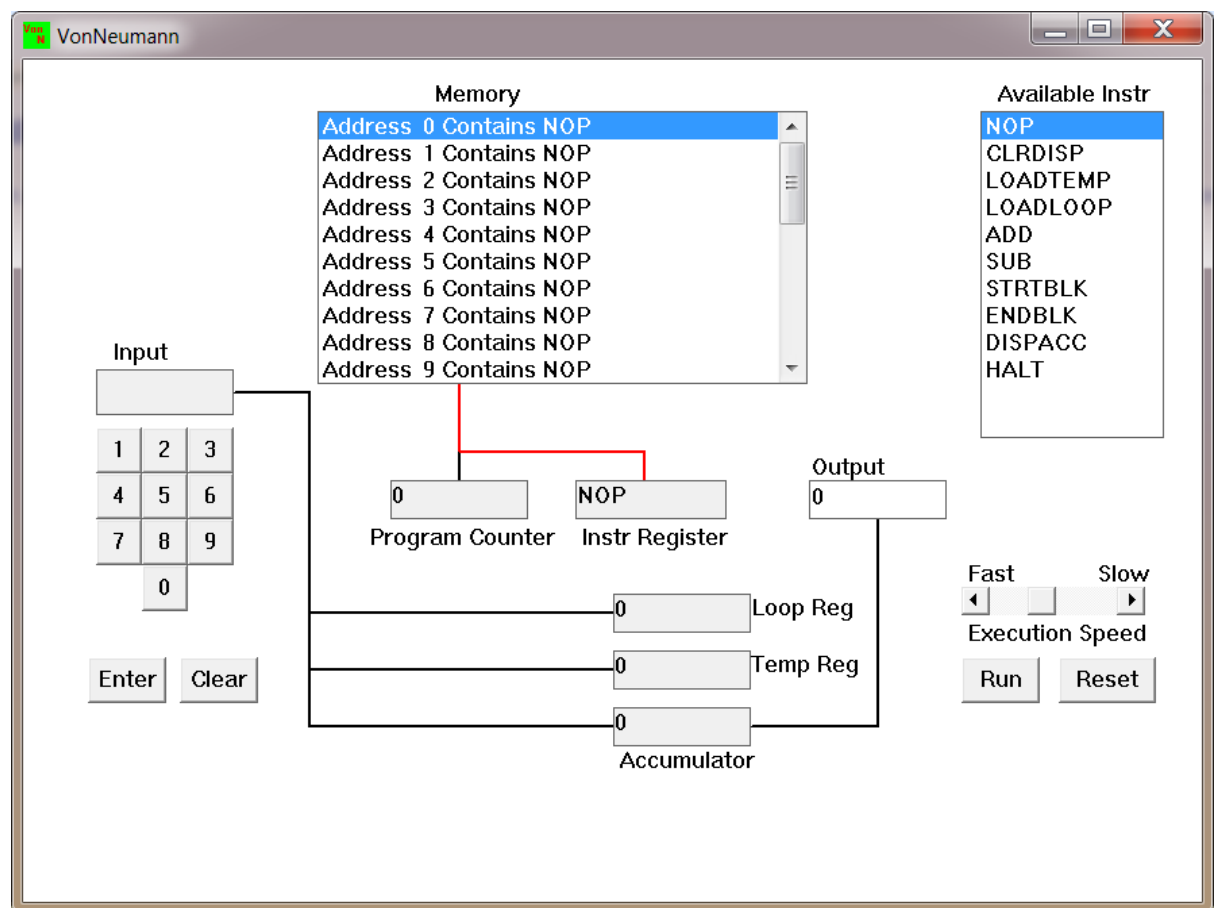Instruction:

Complete all questions in 2 hours.

Name: Apeksha Khanal

ID: np03cs4s230070

1. Von Neumann-Simulator. This program simulates a very simple computer with the von Neumann architecture.

   a. Download the von Neumann Simulator (VonNeumann.exe) program from google classroom Week 5 folder. Save it in your Documents folder and run it. You will see a window similar to this:

The simulator has a small program memory area which is available for programming. To enter your program instructions simply click on the "Available" instruction on the list on the right and then click on the "Memory" location you wish to put it in.

This simulator understands only the following ten instructions:

NOP	No Operation, i.e. do nothing.
LOADTEMP	Get a number from the keypad, completed by the Enter key, into the Temporary Register.
LOADLOOP	Get a number from the keypad, completed by the Enter key, into the Loop Register.
CLRDSP	Clear the Display.
ADD	Add the Temporary Register to the Accumulator
SUB	Subtract the Temporary Register from the Accumulator
DISPACC	Display the contents of the Accumulator
STRTBLK	Start of Loop Block
ENDBLK	End of Loop Block
HALT	Halt. Stop Program

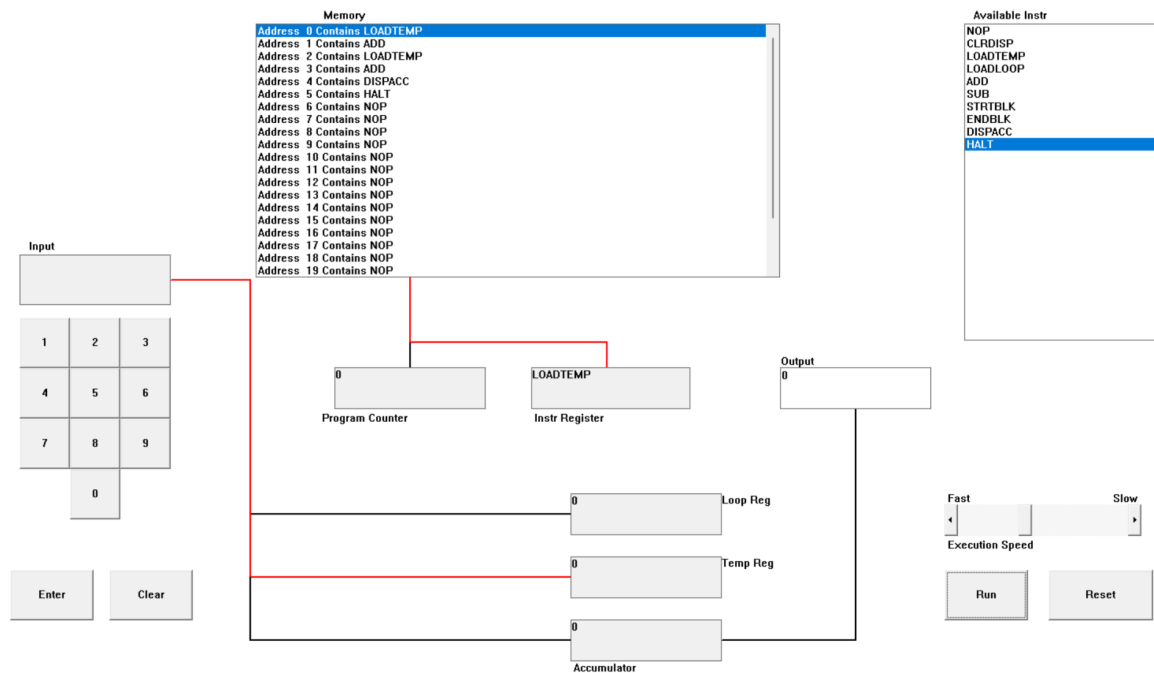b. Load the following program in the memory and explain what the program does?

LOADTEMP

ADD

LOADTEMP

ADD

DISPAAC

HALT

Memory

Address 0 Contains LOADTEMP
Address 1 Contains ADD
Address 2 Contains LOADTEMP
Address 3 Contains ADD
Address 4 Contains DISPACC
Address 5 Contains HALT
Address 6 Contains NOP
Address 7 Contains NOP
Address 8 Contains NOP
Address 9 Contains NOP
Address 10 Contains NOP
Address 11 Contains NOP
Address 12 Contains NOP
Address 13 Contains NOP
Address 14 Contains NOP
Address 15 Contains NOP
Address 16 Contains NOP
Address 17 Contains NOP
Address 18 Contains NOP
Address 19 Contains NOP

Available Instr

NOP
CLRDISP
LOADTEMP
LOADLOOP
ADD
SUB
STRTBLK
ENDBLK
DISPACC
HALT

Input

1	2	3
4	5	6
7	8	9
	0	

Enter   Clear

0

Program Counter

LOADTEMP

Instr Register

Output
0

0 **Loop Reg**

0 **Temp Reg**

0 **Accumulator**

Fast   Slow

Execution Speed

Run   Reset
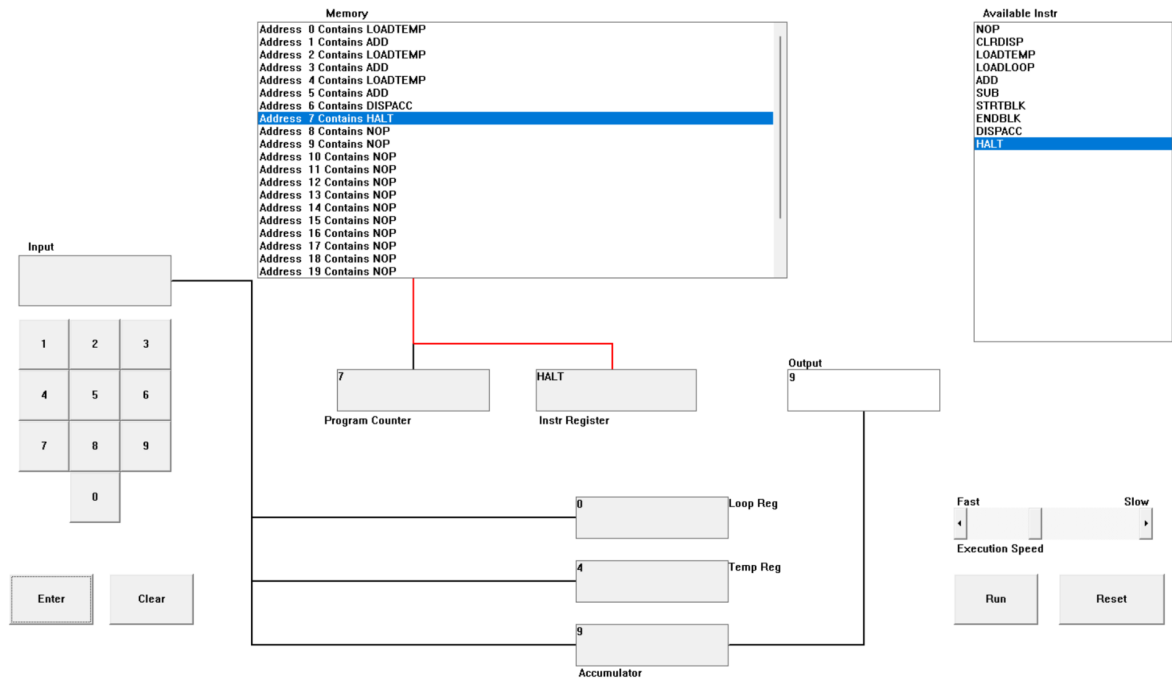
The above program contains Program counter, Accumulator, instruction register, temp register, input, memory and output. When we load the program, the program LOADTEMP takes input from the user and the input value passes through the Accumulator. When we run the program with the input value the place of memory will be at top and the instruction register will have LOADTEMP. When we do add, the two numbers will be added in the program. Then again the input is taken from the user and added. The DISPACC shows the output to the user and halt causes the program to stop.

c. Write the program to add three numbers together and explain how your code works?

Memory

```
Address  0 Contains LOADTEMP
Address  1 Contains ADD
Address  2 Contains LOADTEMP
Address  3 Contains ADD
Address  4 Contains LOADTEMP
Address  5 Contains ADD
Address  6 Contains DISPACC
Address  7 Contains HALT
Address  8 Contains NOP
Address  9 Contains NOP
Address 10 Contains NOP
Address 11 Contains NOP
Address 12 Contains NOP
Address 13 Contains NOP
Address 14 Contains NOP
Address 15 Contains NOP
Address 16 Contains NOP
Address 17 Contains NOP
Address 18 Contains NOP
Address 19 Contains NOP
```

Available Instr

```
NOP
CLRDISP
LOADTEMP
LOADLOOP
ADD
SUB
STRTBLK
ENDBLK
DISPACC
HALT
```

Input

1	2	3
4	5	6
7	8	9
	0	

Enter   Clear

Program Counter: 7

Instr Register: HALT

Output: 9

Loop Reg: 0

Temp Reg: 4

Accumulator: 9

Fast —— Slow
Execution Speed

Run   Reset

In the above program, I have done LOADTEMP, ADD, LOADTEMP, ADD, LOADTEMP, ADD, DISPACC and HALT. As we know how all of these work, I have given the input 2,3 and 4 through LOADTEMP and it simply loads and the function ADD adds the three numbers and DISPACC has displayed the output 9. At last the program is stopped by HALT.
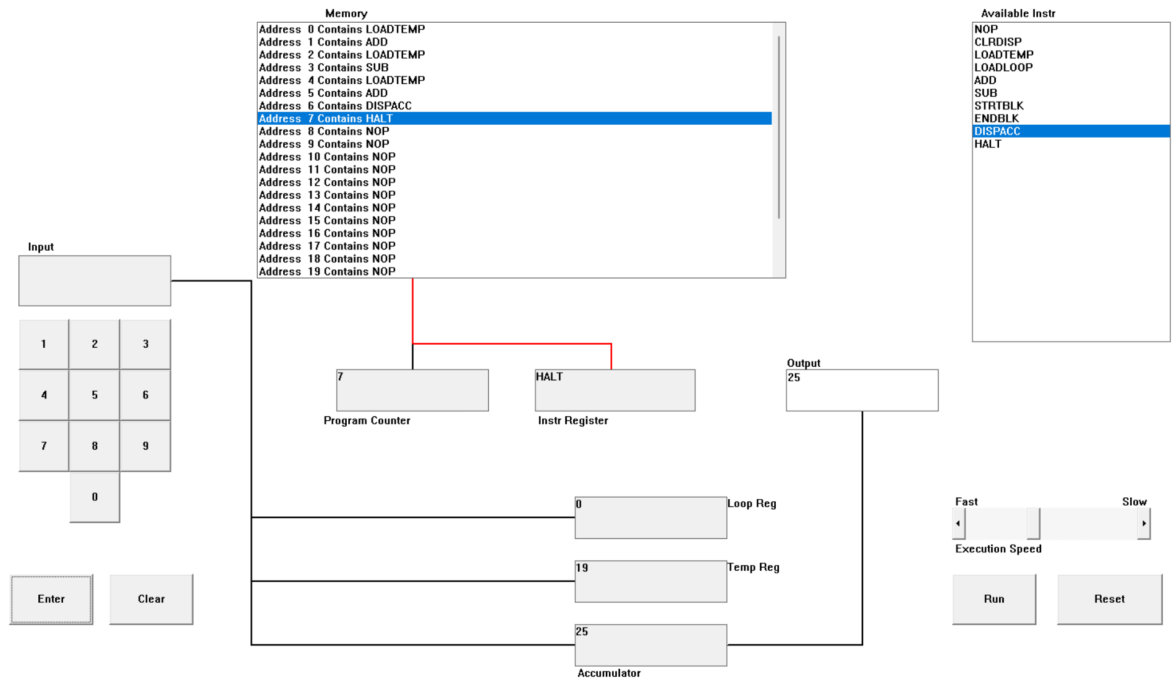
d. Write the program to perform
   7+3-9
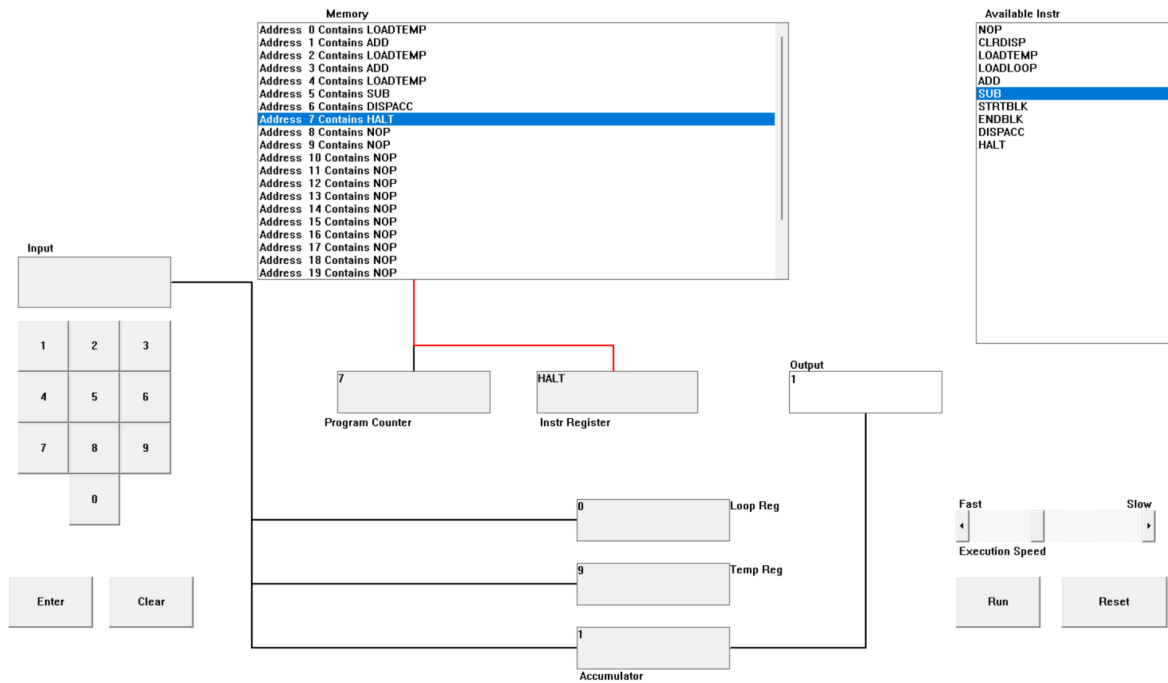
In the above program, I have done LOADTEMP, ADD, LOADTEMP, ADD, LOADTEMP, SUB, DISPACC and HALT. At first the user input 7 is entered through LOADTEMP and then 3 is entered. The add function adds these numbers which become 10. Then, another value 9 is loaded which is then subtracted from 10. The output becomes 1 and HALT stops the program.
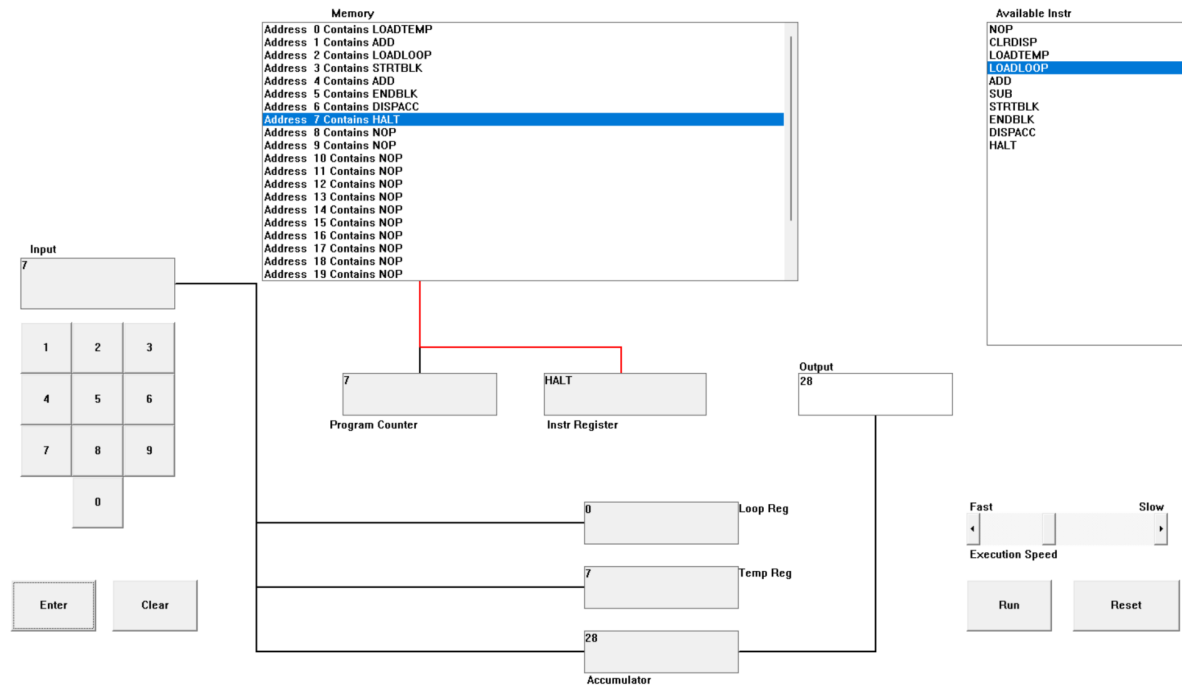
9+3-7
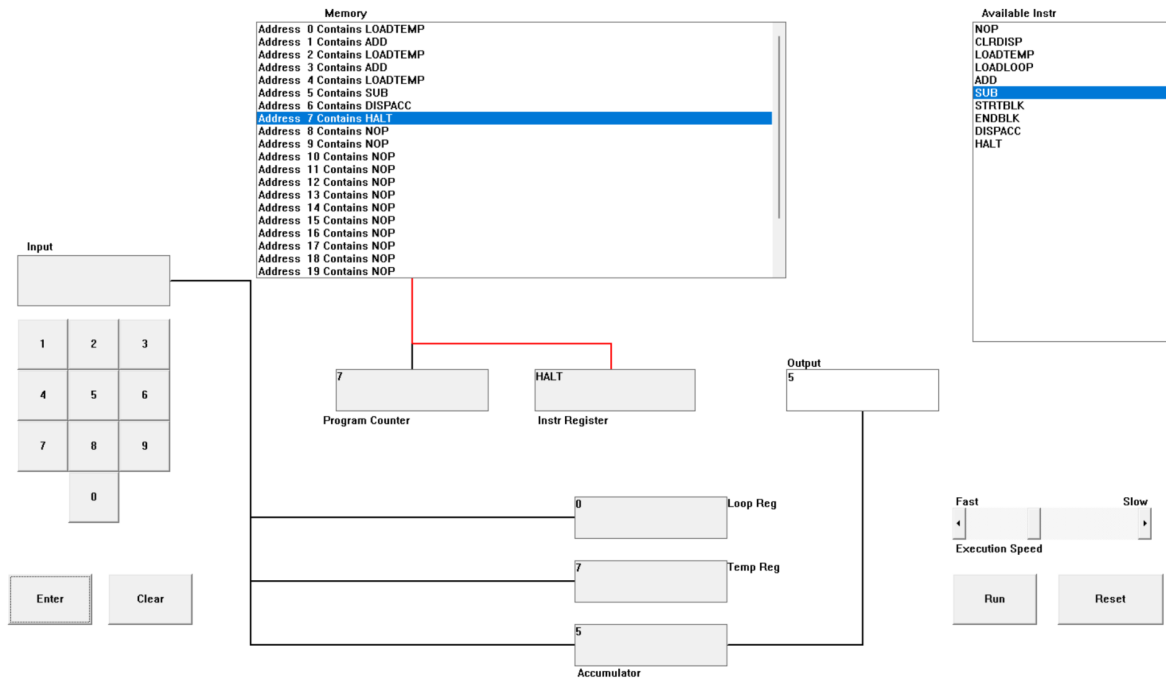
In the above program, I have done LOADTEMP, ADD, LOADTEMP, ADD, LOADTEMP, SUB, DISPACC and HALT. At first the input 9 is given through loadtemp and the second input is 3. The ADD function simply adds these numbers and it becomes 12. Now another input 7 is given and the SUB function subtracts 7 from 12. The output becomes 5 and HALT stops the program.

13-7+19

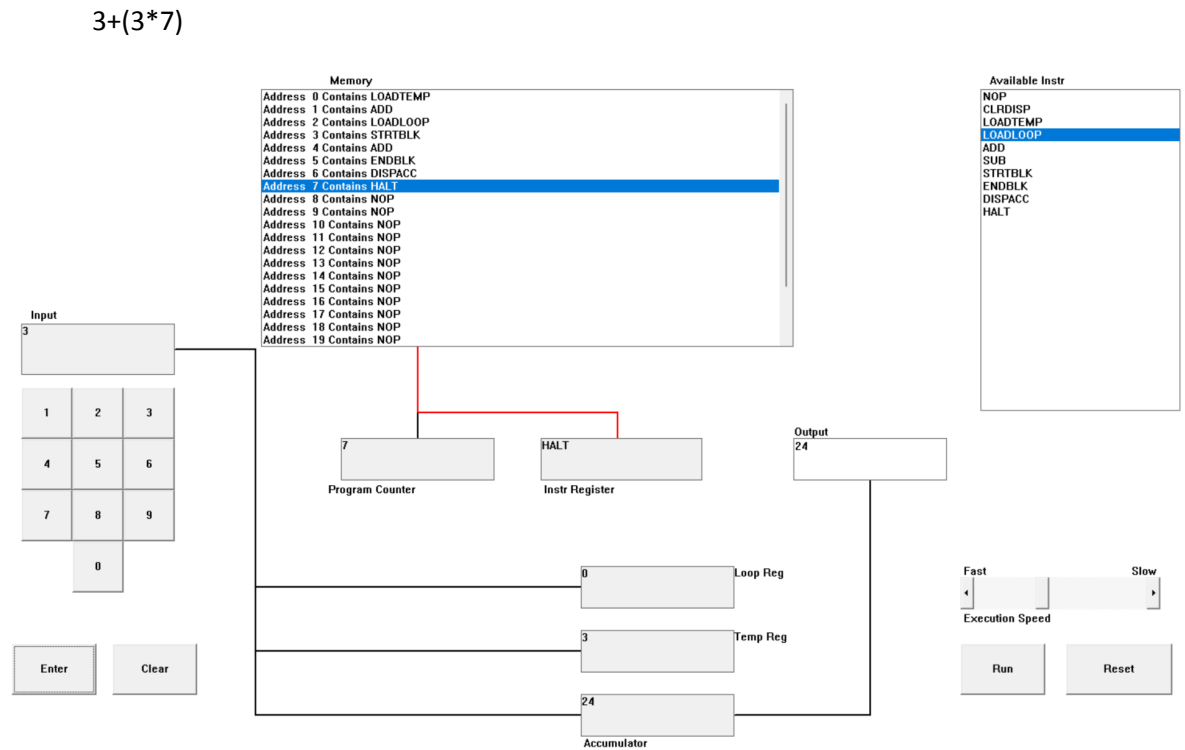In the above program I have done LOADTEMP, ADD, LOADTEMP, SUB, LOADTEMP, ADD, DISPACC and HALT. At first, input 13 is entered through LOADTEMP and then the second input 7 is entered. The SUB function simply subtracts 7 from 13 and it becomes 6. Then, the third input 19 is entered. Now the ADD function adds 6 and 19 and the output becomes 25. The output is displayed through DISPACC and HALT stops the program.

e. Write a program to perform
    7+(7*3)

Memory

Address 0	Contains LOADTEMP
Address 1	Contains ADD
Address 2	Contains LOADLOOP
Address 3	Contains STRTBLK
Address 4	Contains ADD
Address 5	Contains ENDBLK
Address 6	Contains DISPACC
Address 7	Contains HALT
Address 8	Contains NOP
Address 9	Contains NOP
Address 10	Contains NOP
Address 11	Contains NOP
Address 12	Contains NOP
Address 13	Contains NOP
Address 14	Contains NOP
Address 15	Contains NOP
Address 16	Contains NOP
Address 17	Contains NOP
Address 18	Contains NOP
Address 19	Contains NOP

Available Instr

NOP
CLRDISP
LOADTEMP
LOADLOOP
ADD
SUB
STRTBLK
ENDBLK
DISPACC
HALT

Input
7

1	2	3
4	5	6
7	8	9
	0	

Enter   Clear

Program Counter: 7

Instr Register: HALT

Output: 28

Loop Reg: 0

Temp Reg: 7

Accumulator: 28

Fast ◄ ► Slow
Execution Speed

Run   Reset

In the above program, I have done LOADTEMP, ADD, LOADLOOP, STRTBLK, ADD, ENDBLK, DISPACC and HALT. At first, after running the program, the first input 7 is loaded through LOADTEMP. Then, the value 3 is entered for a loop. After that, the input 7 is entered. The output 28 is shown by the DISPACC and HALT simply ends the program.

3+(3*7)

Memory

```
Address  0 Contains LOADTEMP
Address  1 Contains ADD
Address  2 Contains LOADLOOP
Address  3 Contains STRTBLK
Address  4 Contains ADD
Address  5 Contains ENDBLK
Address  6 Contains DISPACC
Address  7 Contains HALT
Address  8 Contains NOP
Address  9 Contains NOP
Address 10 Contains NOP
Address 11 Contains NOP
Address 12 Contains NOP
Address 13 Contains NOP
Address 14 Contains NOP
Address 15 Contains NOP
Address 16 Contains NOP
Address 17 Contains NOP
Address 18 Contains NOP
Address 19 Contains NOP
```

Available Instr

```
NOP
CLRDISP
LOADTEMP
LOADLOOP
ADD
SUB
STRTBLK
ENDBLK
DISPACC
HALT
```

Input
3

1	2	3
4	5	6
7	8	9
	0	

Enter   **Clear**

7 — Program Counter

HALT — Instr Register

Output
24

0 — Loop Reg

3 — Temp Reg

24 — Accumulator

Fast ◄———— Slow
Execution Speed

Run   **Reset**

In the above program, I have done LOADTEMP, ADD, LOADLOOP, STRTBLK, ADD, ENDBLK, DISPACC and Halt. The first input is 3 which is entered through LOADTEMP. Then, for the loop, input 7 is entered. The third input is 3 and the output becomes 24. At last HALT ends the program.

f.   Write a program to add the first 10 natural numbers.
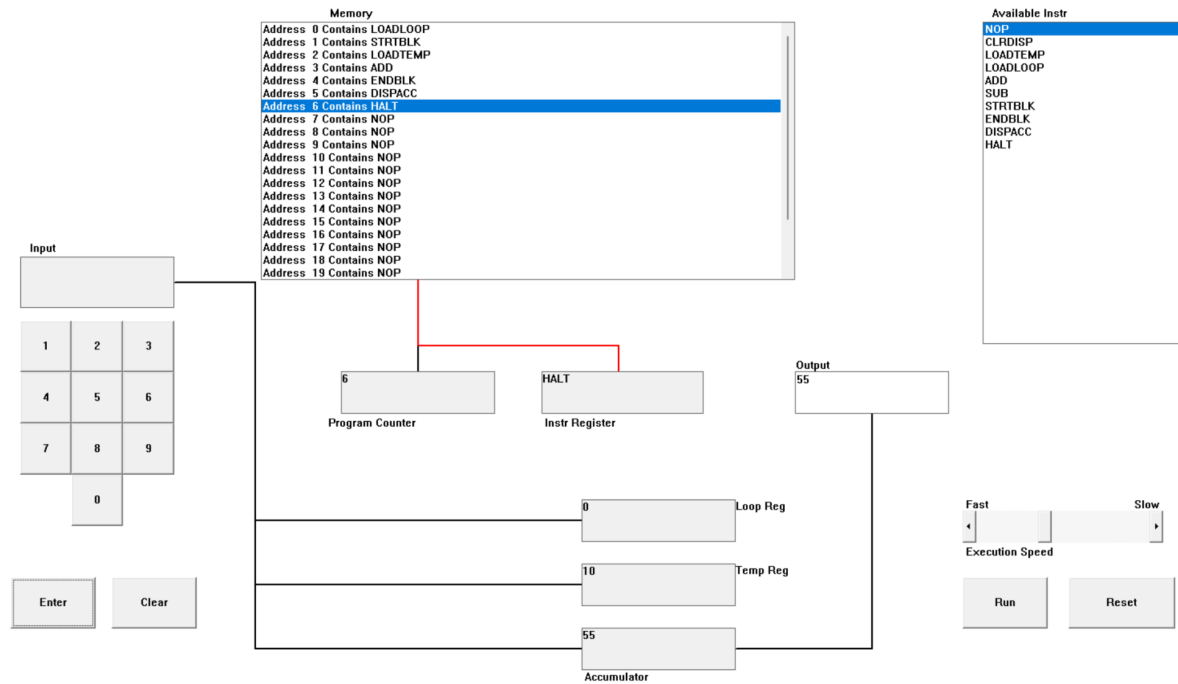
In the above program, I have done LOADLOOP, STRTBLK, LOADTEMP, ADD, ENDBLK, DISPACC and HALT. For the loop I have entered 10 so that I can enter all the 10 natural numbers one by one. Then, I have entered all the natural numbers one after another and the ADD function has added it. The output is 55 which is displayed and at last this program is ended by HALT.