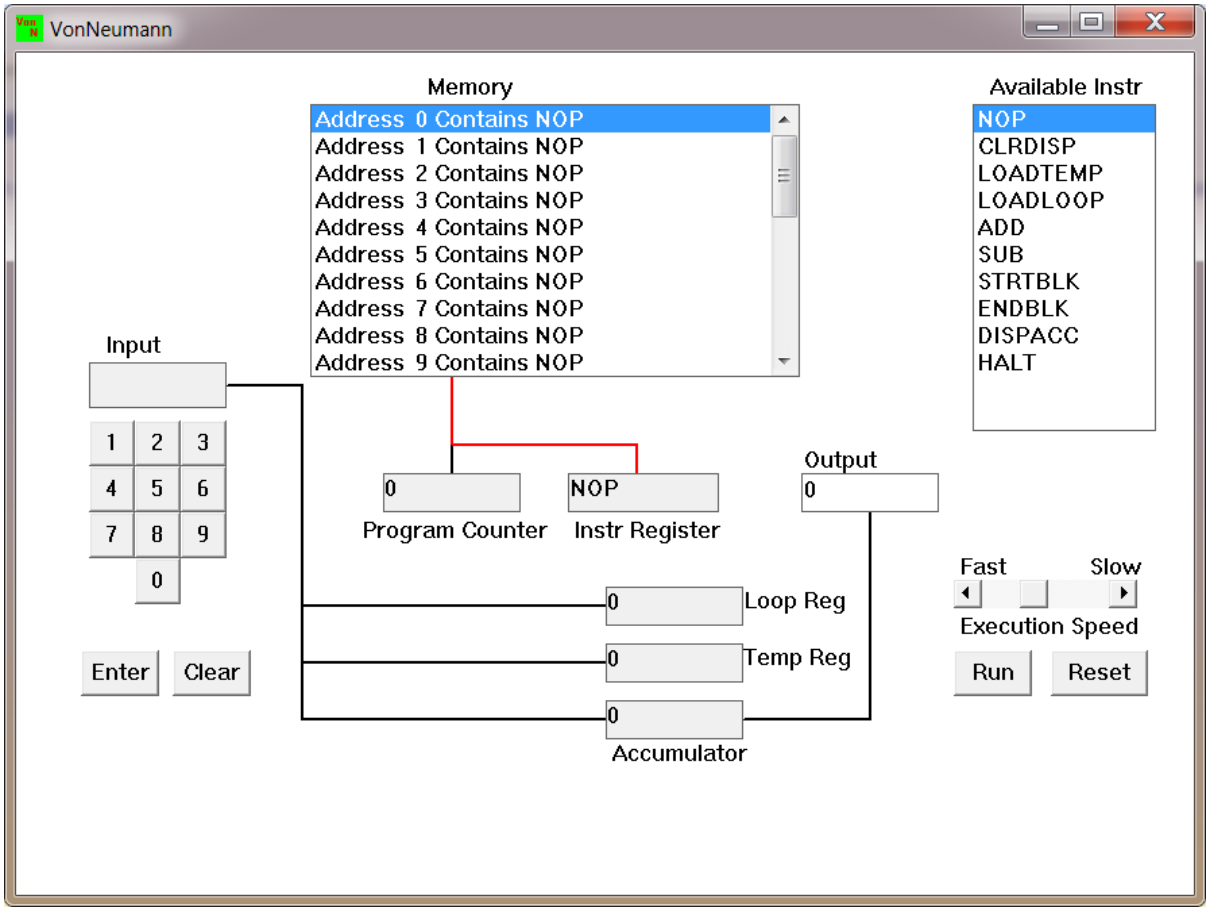
Instruction:

Complete all questions in 2 hours.

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 in the Week-7 folder. Save it in your Documents folder and run it. You will see a window similar to this:



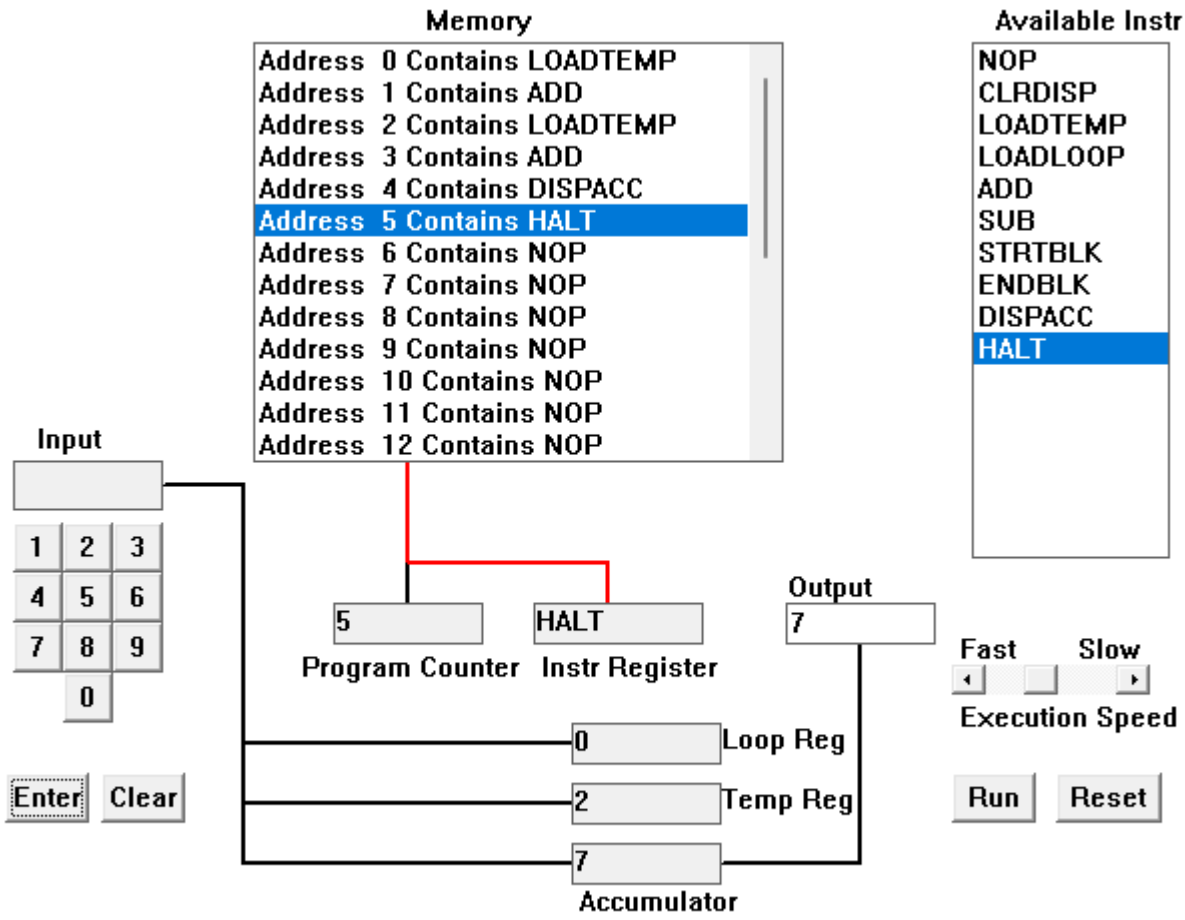
The simulator has a small program memory area which is available for programming. To enter your program instructions simply click on the “Available” instruction on the list on the right and then click on the “Memory” location you wish to put it in.

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.
CLRDISP	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 does the program does?

```
LOADTEMP
ADD
LOADTEMP
ADD
DISPAAC
HALT
```



= LOADTEMP (5)

ADD

LOADTEMP (2)

ADD

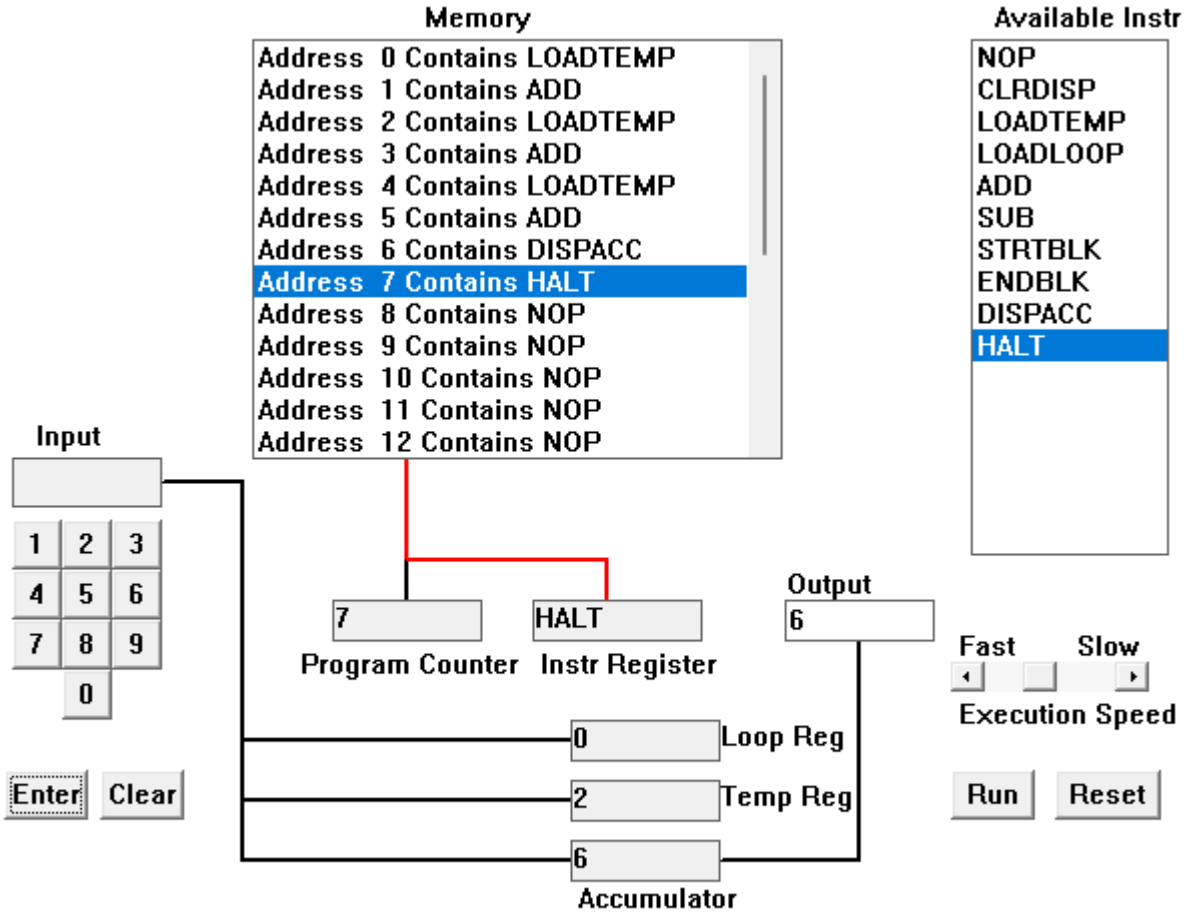
DISPACC

HALT

= In the program, the LOADTEMP instruction asks for an input where the entered input goes and gets stored in the temp register. The ADD instruction goes to the accumulator and does the addition with the next input asked by another LOADTEMP. Therefore, after the both

inputs are added, the DISPACC displays the output and the HALT instructions stops the program from running.

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



= In the given program, to add the three numbers we have used three similar instructions each for LOADTEMP and ADD. The LOADTEMP instructions asks for an input store it in the temp register and the ADD instructions adds the inputs stored in an accumulator. The DISPACC displays the output which was a result in an accumulator and the HALT instruction stops the program from running after an output is displayed.

d. Write the program to perform

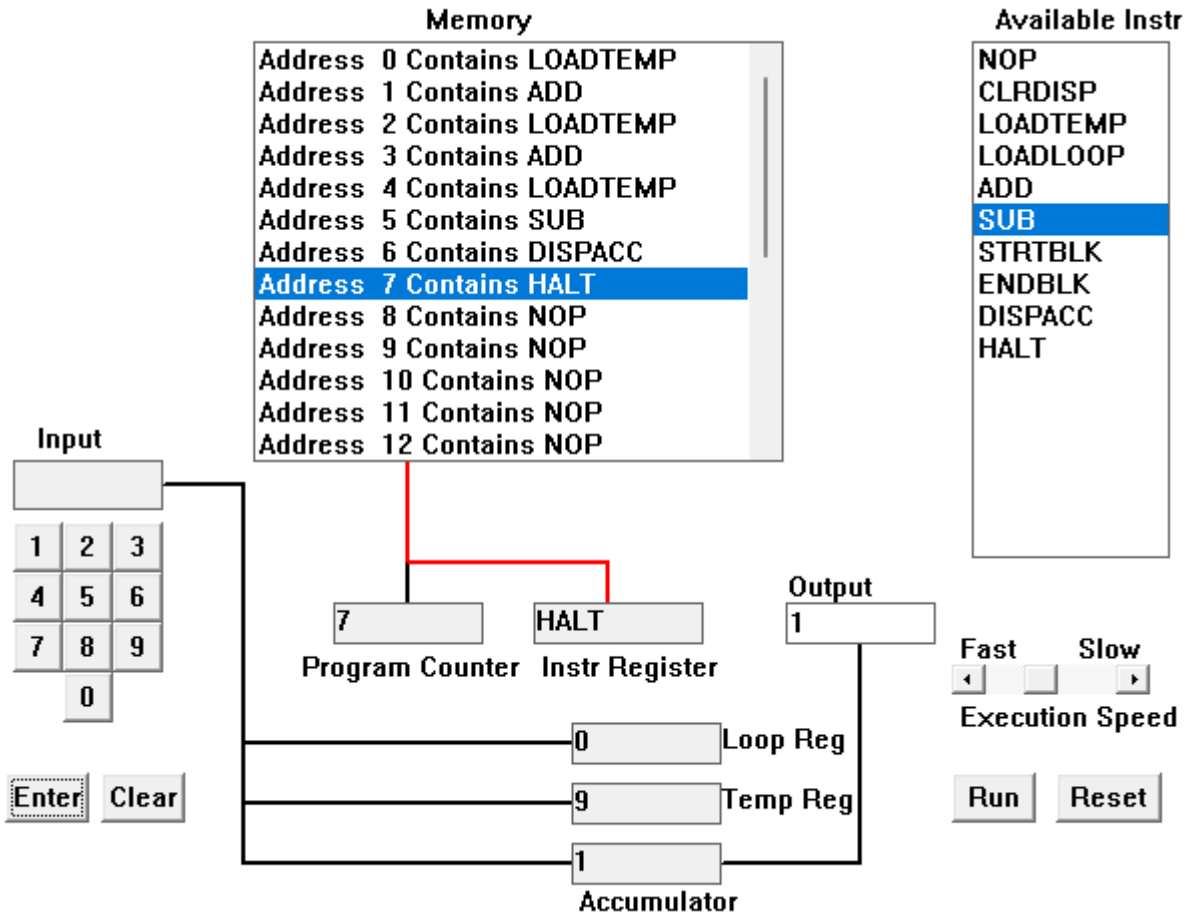
7+3-9

-9+3-7

13-7+19

List your code here.

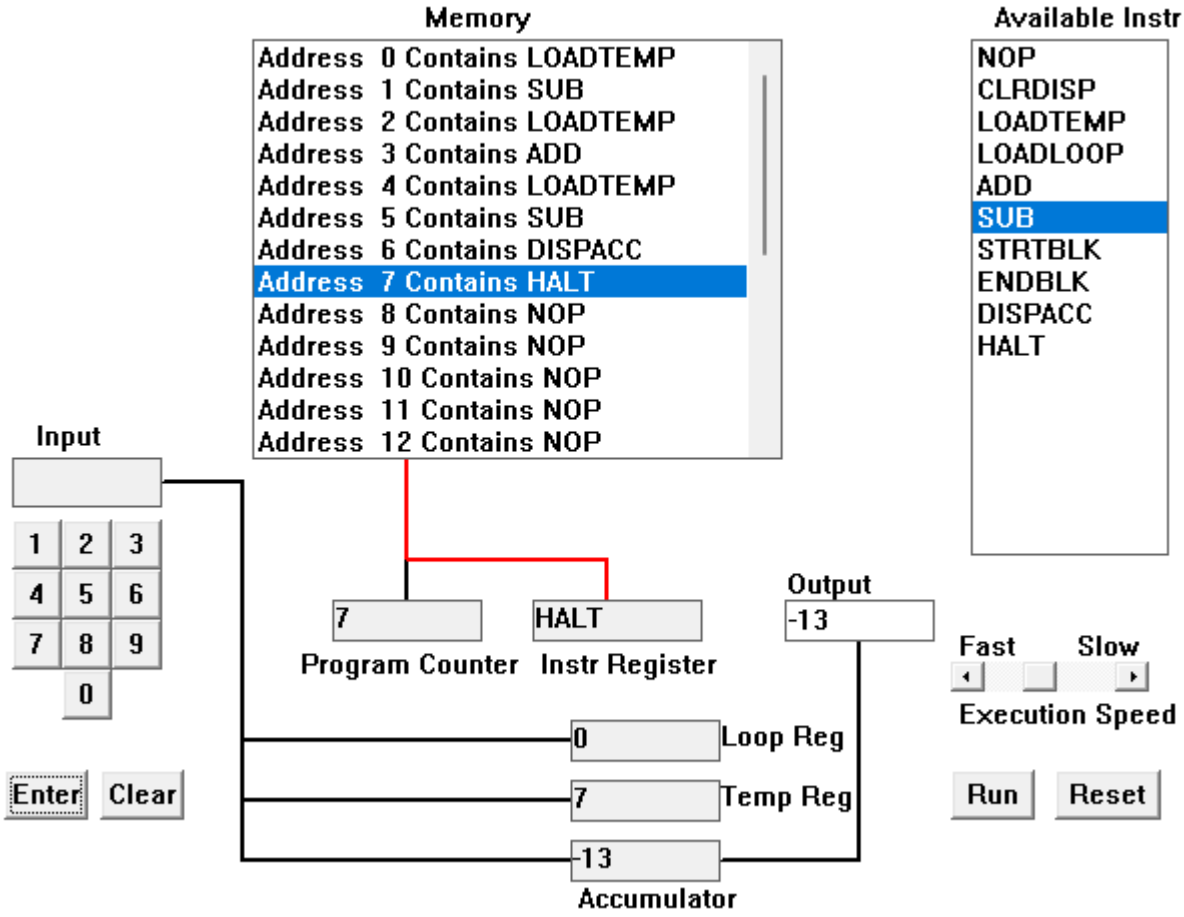
7+3-9



Code:

LOADTEMP (7)
ADD
LOADTEMP (3)
ADD
LOADTEMP (9)
SUB
DISPACC
HALT

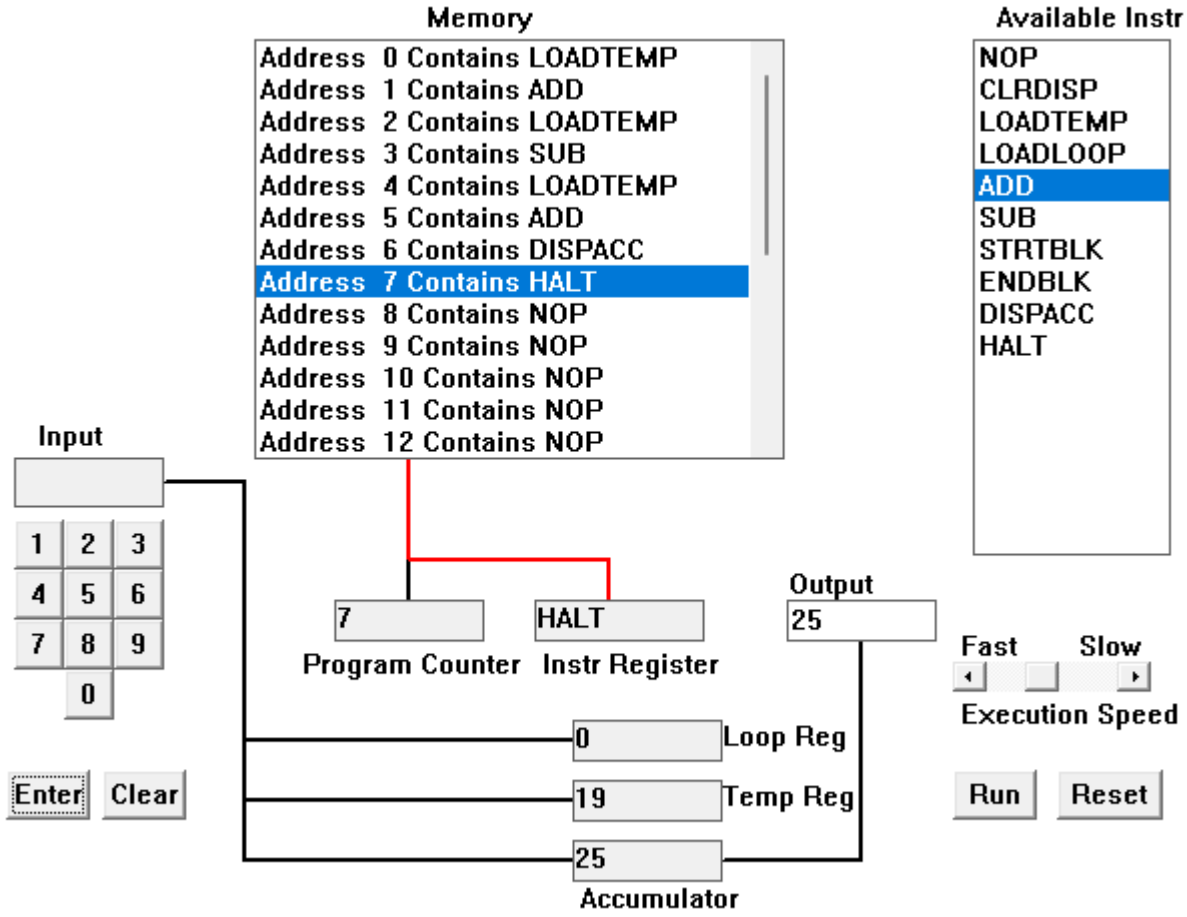
-9+3-7



Code:

LOADTEMP (9)
SUB
LOADTEMP (3)
ADD
LOADTEMP (7)
SUB
DISPACC
HALT

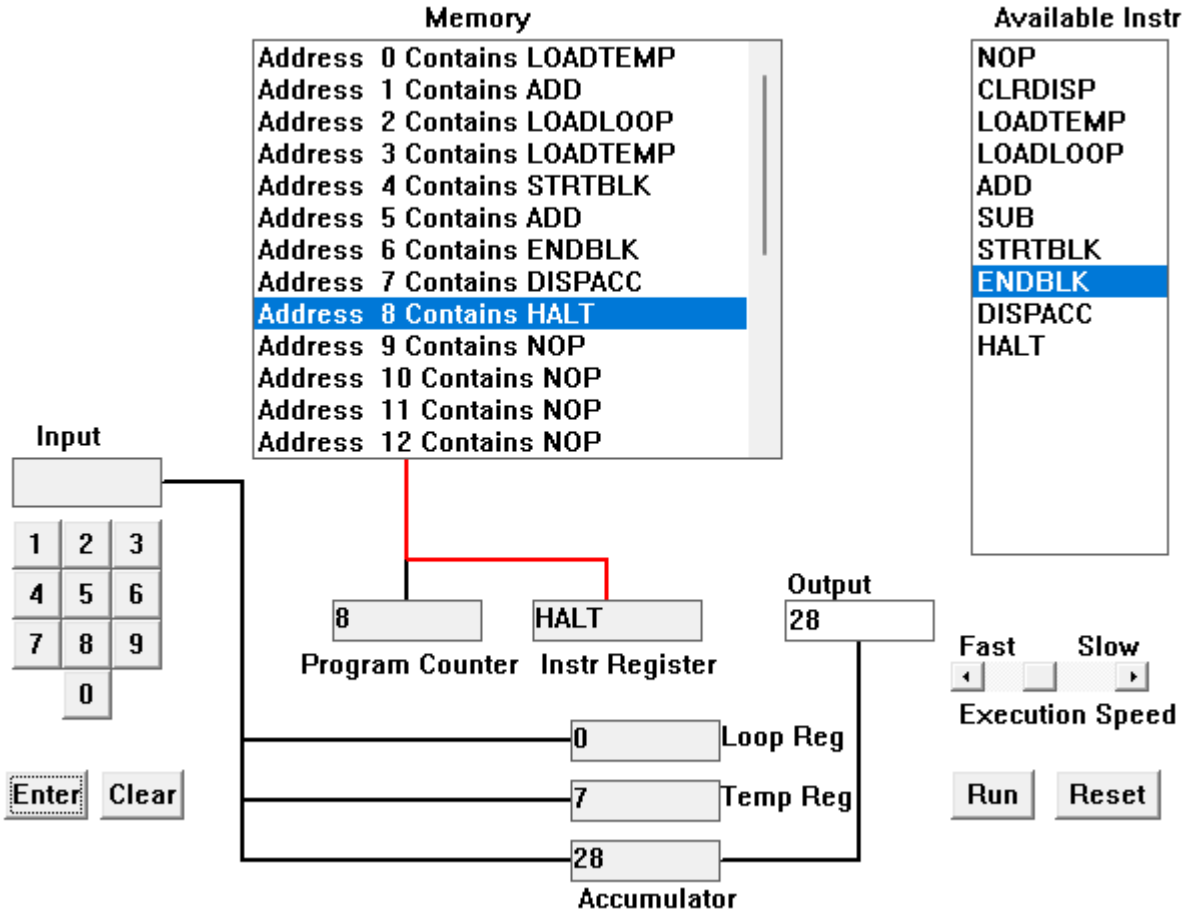
13-7+19



Code:
LOADTEMP (13)
ADD
LOADTEMP (7)
SUB
LOADTEMP (19)
ADD
DISPACC
HALT

e. Write a program to perform
7+(7*3)
3+(3*7)
List your code here.

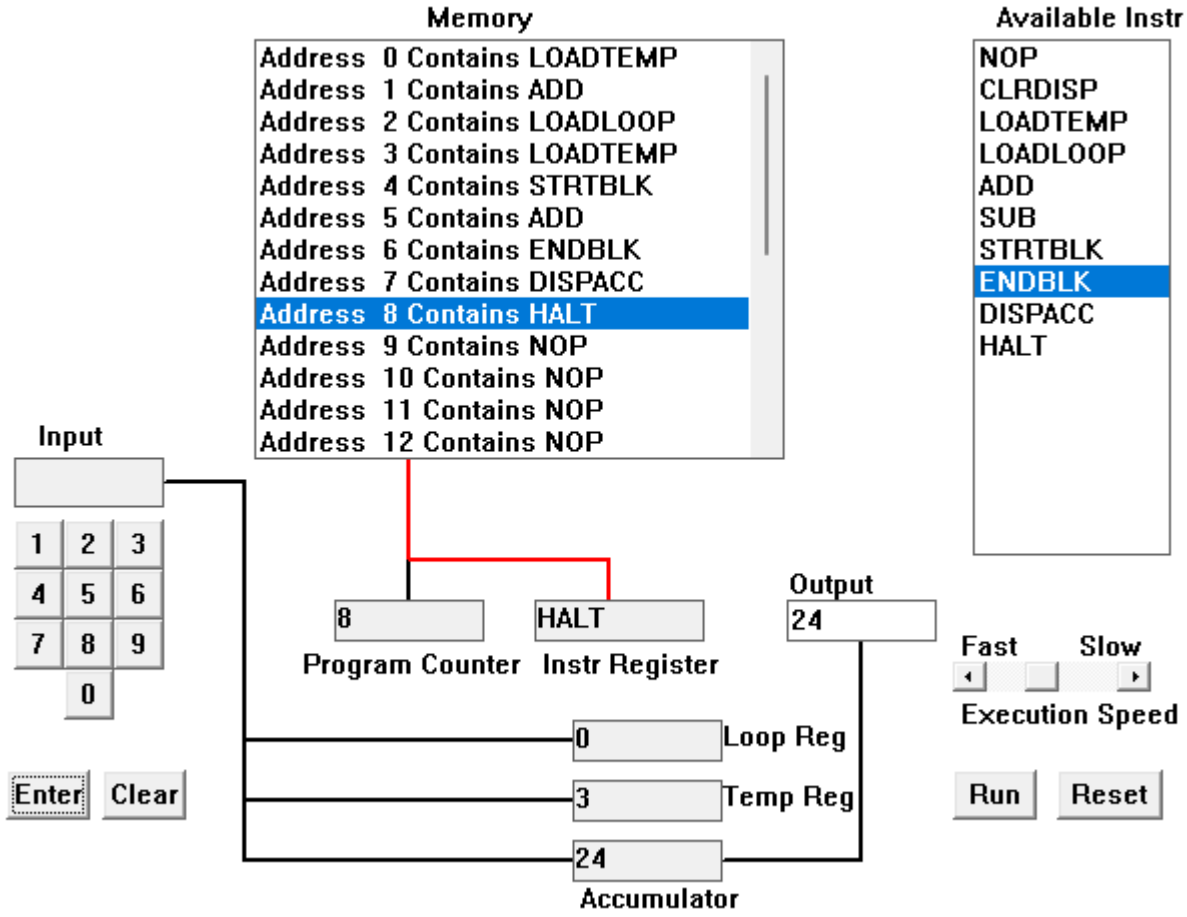
7+(7*3)



Code:

```
LOADTEMP (7)
ADD
LOADLOOP (3)
LOADTEMP (7)
STRTBLK
ADD
ENDBLK
DISPACC
HALT
```

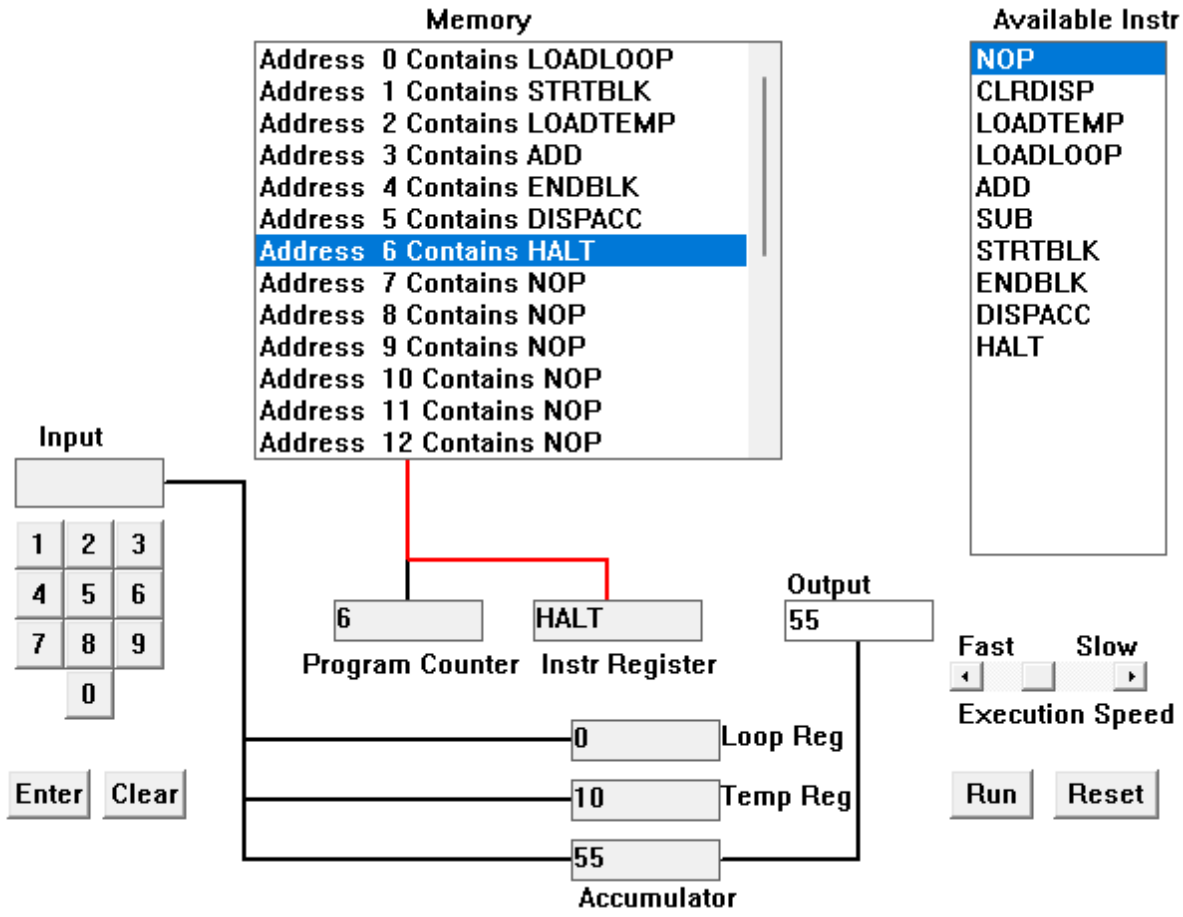
3+(3*7)



Code:

```
LOADTEMP (3)
ADD
LOADLOOP (7)
LOADTEMP (3)
STRTBLK
ADD
ENDBLK
DISPACC
HALT
```

f. Write a program to add first 10 natural numbers.


Memory
Address 0 Contains LOADLOOP
Address 1 Contains STRTBLK
Address 2 Contains LOADTEMP
Address 3 Contains ADD
Address 4 Contains ENDBLK
Address 5 Contains DISPACC
Address 6 Contains HALT
Address 7 Contains NOP
Address 8 Contains NOP
Address 9 Contains NOP
Address 10 Contains NOP
Address 11 Contains NOP
Address 12 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
6
Program Counter
HALT
Instr Register
Output
55
Fast
Slow
Execution Speed
0
Loop Reg
10
Temp Reg
55
Accumulator
Run
Reset