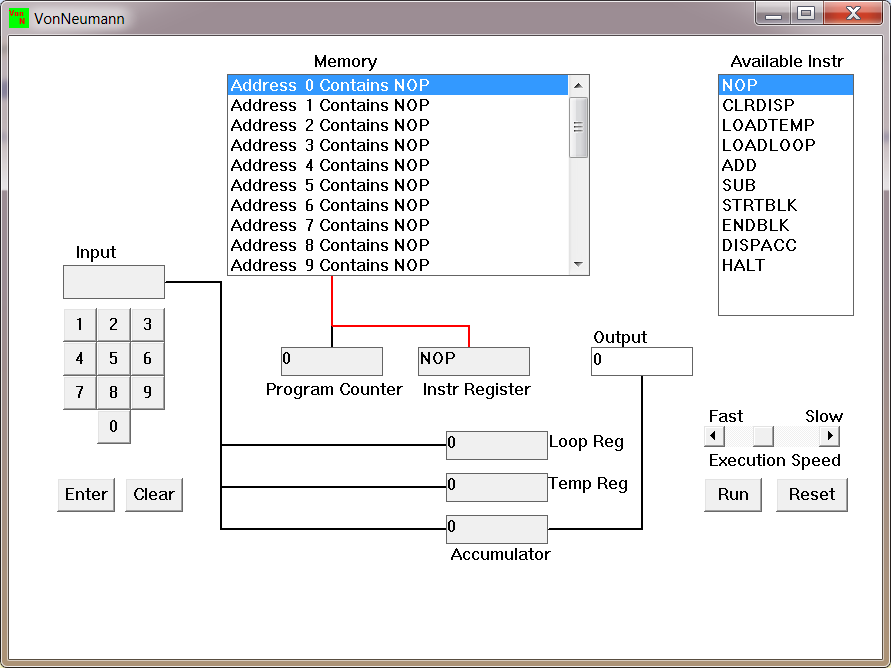
Instruction:

Complete all questions in 2 hour.

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

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

LOADTEMP(69) accepts the User input

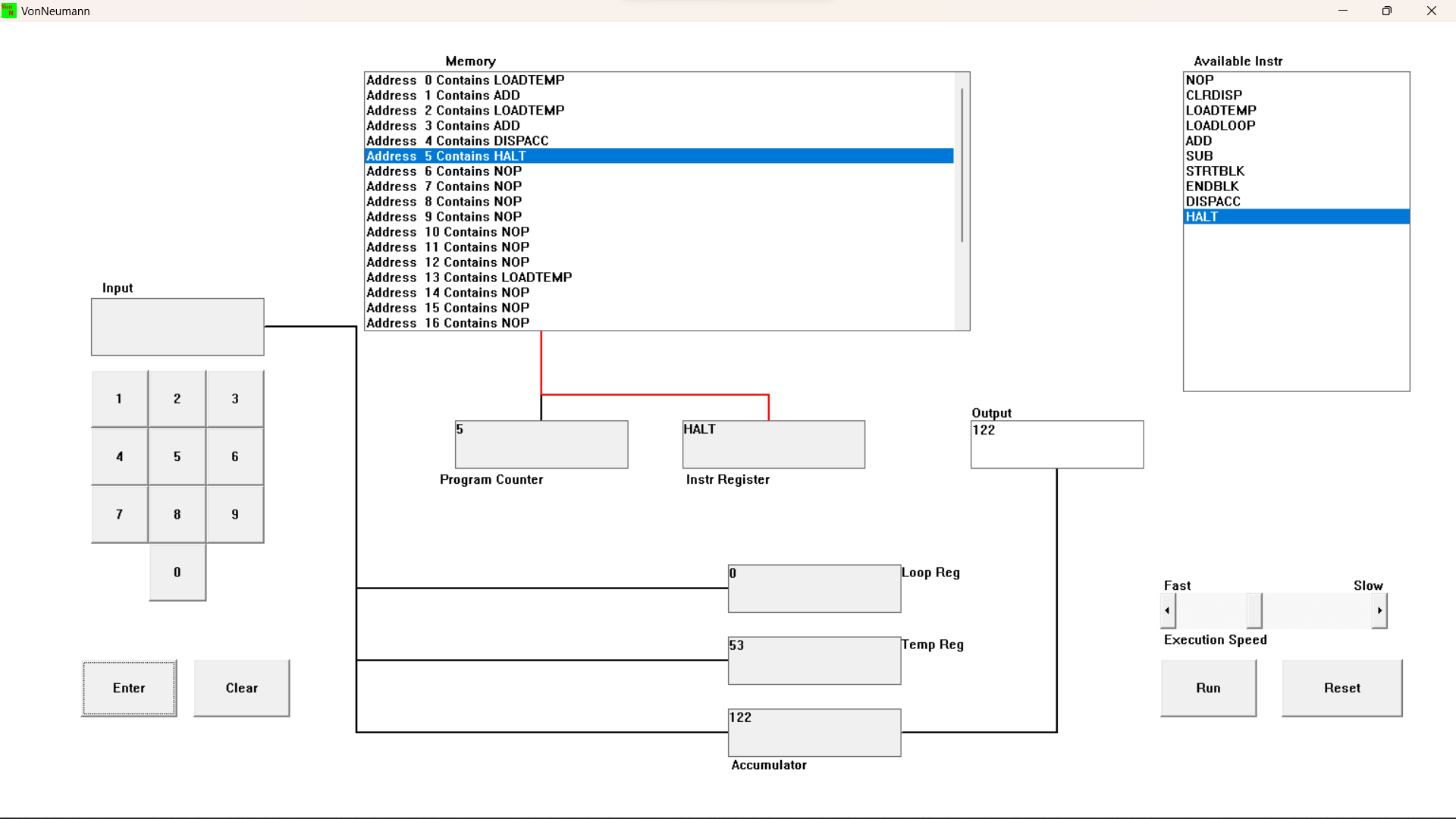
ADD(69) adds the user input to its initial value which is ‘0’

LOADTEMP(53) second user input

ADD(122) add the user input to its temporary registered value which is ‘53’

DISPAAC(122) displays the output

HALT



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

Diagram

Description automatically generated

LOADTEMP(2) { User’s input}

ADD(2) { adds the user input to its initial value which is ‘0’ in the accumulator }

LOADTEMP(3) { second user input}

ADD(5) { add the user input to its temporary registered value which is ‘2’ }

LOADTEMP(3) second user input

ADD(8) add the user input to its temporary registered value which is ‘3 ’

DISPAAC(8) displays the final data

HALT

* 1. Write the program to perform

7+3-9

Diagram

Description automatically generated

LOADTEMP(7)

ADD(7)

LOADTEMP(3)

ADD(10)

LOADTEMP(9)

SUB(1)

DISPAAC(1)

HALT

-9+3-7

Diagram

Description automatically generated

LOADTEMP(9)

SUB(9)

LOADTEMP(3)

ADD(-6)

LOADTEMP(-7)

SUB(-13)

DISPAAC(-13)

HALT

13-7+19

Diagram

Description automatically generated with low confidence

LOADTEMP(13)

ADD(13)

LOADTEMP(-7)

SUB(6)

LOADTEMP(19)

ADD(25)

DISPAAC(25)

HALT

* 1. Write a program to perform

7+(7\*3)

LOADTEMP = 7

ADD = 0+7

LOADTEMP = 7

LOADLOOP = 3

STRTBLK

ADD = 7+7

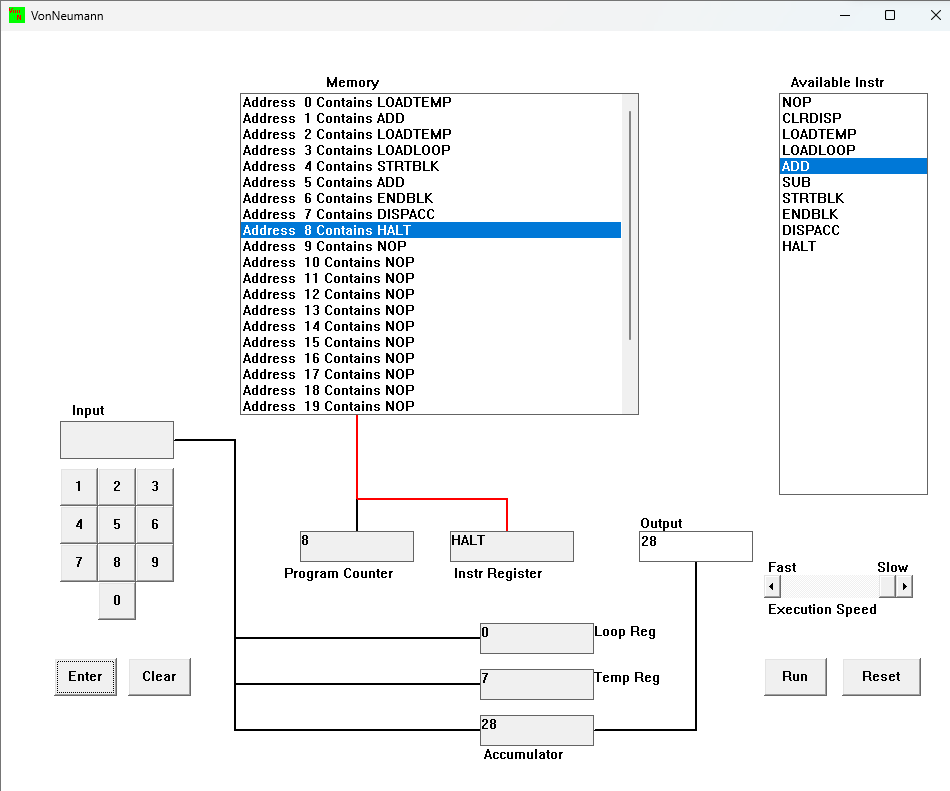
ADD = 14+7

ADD= 21+7

ENDBLK

DISPACC= 28

HALT



3+(3\*7)

LOADTEMP

ADD

LOADTEMP

LOADLOOP

STRTBLK

ADD

ENDBLK

DISPACC

HALT

Diagram

Description automatically generated

* 1. Write a program to add first 10 natural numbers.

