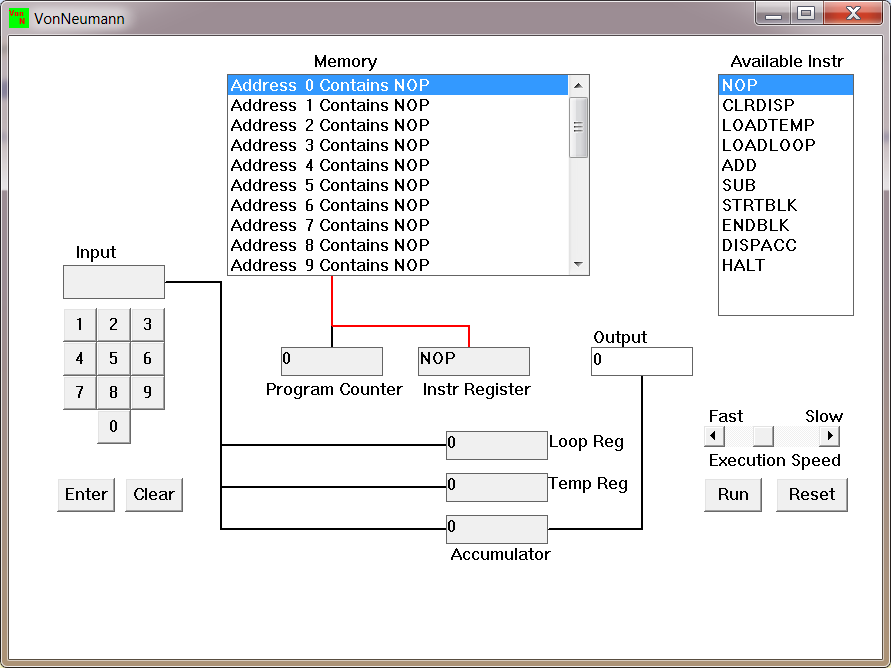
Instruction:

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 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. |
| 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

ADD

LOADTEMP

ADD

DISPAAC

HALT

Diagram

Description automatically generated

This program first stores the positive number on temporary register which is due to the LOADTEMP command or instruction from the instruction register. The ADD instruction after LOADTEMP determines that the sign of the entered number is positive. After that, it also does the same again, which is it stores the positive number again in the temporary register due to the LOADTEMP Instruction. Then it adds the two entered numbers together and stores it in the accumulator. After that, it displays the value stored at the accumulator by DISPACC instruction. Finally, it ends the program which is due to the HALT instruction. Thus, the program does the following operations to add two positive numbers together.

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

Answer: The program is given below to add three numbers together.

Step1: LOADTEMP

Step2: ADD

Step3: LOADTEMP

Step4: ADD

Step5: LOADTEMP

Step6: ADD

Step7: DISPACC

Step8: HALT

Explanation: first, we load the first number in the program by using LOADTEMP and we assign the sign of that number, which is ADD, by using ADD. After that we load the second number in the program by using LOADTEMP and assign that number ADD sign and follow the exact same for third number by using LOADTEMP to load the third number and ADD to add all three numbers together.

Thus, the code works in this way.

* 1. Write the program to perform

7+3-9

Step1: LOADTEMP

Step2: ADD

Step3: LOADTEMP

Step4: ADD

Step5: LOADTEMP

Step6: SUB

Step7: DISPACC

Step8: HALT

Diagram

Description automatically generated

-9+3-7

Step1: LOADTEMP

Step2: ADD

Step3: LOADTEMP

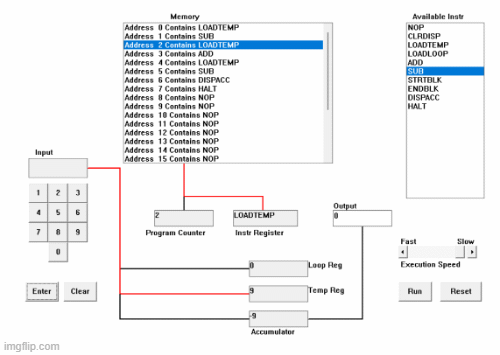
Step4: ADD

Step5: LOADTEMP

Step6: SUB

Step7: DISPACC

Step8: HALT



13-7+19

Step1: LOADTEMP

Step2: ADD

Step3: LOADTEMP

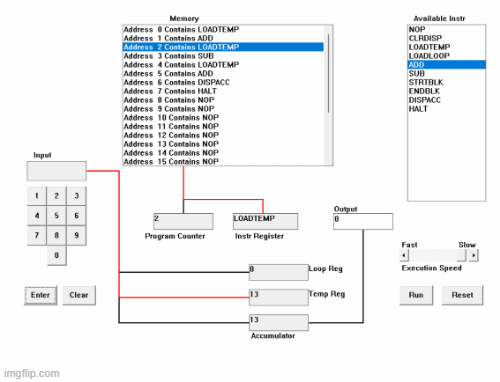
Step4: ADD

Step5: LOADTEMP

Step6: SUB

Step7: DISPACC

Step8: HALT



List your code here.

* 1. Write a program to perform

7+(7\*3)

Step1: LOADTEMP

Step2: ADD

Step3: LOADLOOP

Step4: STRTBLK

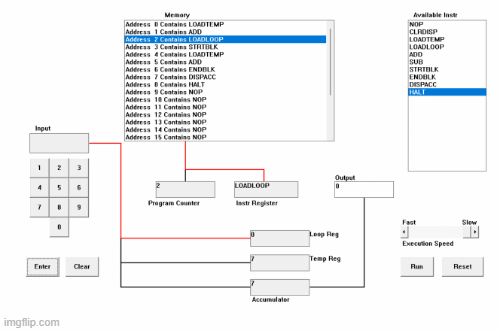
Step5: LOADTEMP

Step6: ADD

Step7: ENDBLK

Step8: DISPACC

Step9: HALT



3+(3\*7)

Step1: LOADTEMP

Step2: ADD

Step3: LOADLOOP

Step4: STRTBLK

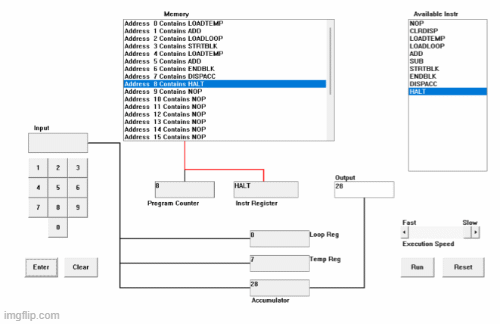
Step5: LOADTEMP

Step6: ADD

Step7: ENDBLK

Step8: DISPACC

Step9: HALT



List your code here.

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

Step1: LOADLOOP

Step2: STRTBLK

Step3: LOADTEMP

Step4: ADD

Step5: ENDBLK

Step6: DISPACC

Step7: HALT

