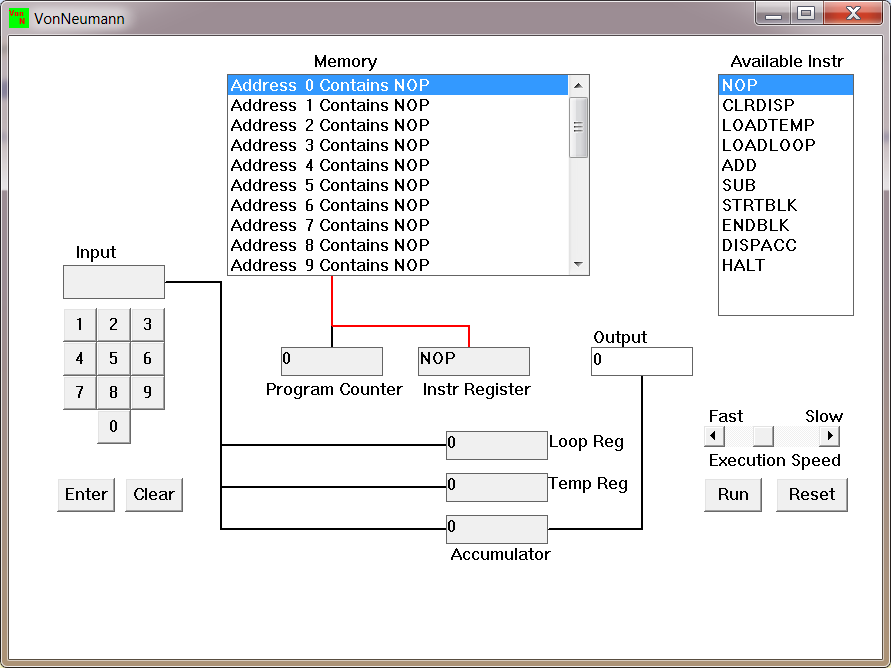
# 4CS015 Fundamentals of Computing – Workshop-6

**Workshop tasks:**

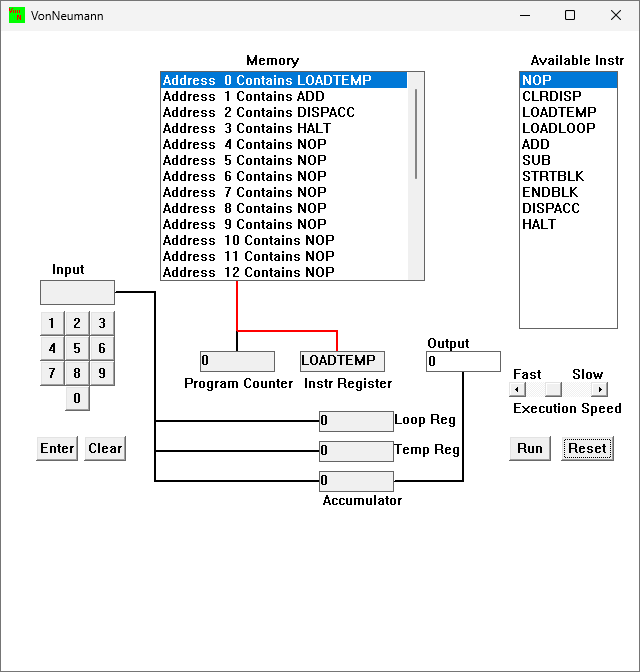
1. Von Neumann Simulator. This program simulates a very simple computer with the von Neumann architecture.
   1. Download the von Neumann Simulator (VonNeumann.exe) program from WOLF in the Week 5 folder. Save it in your Documents folder and run it. You will see a window similar to this:

  
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.

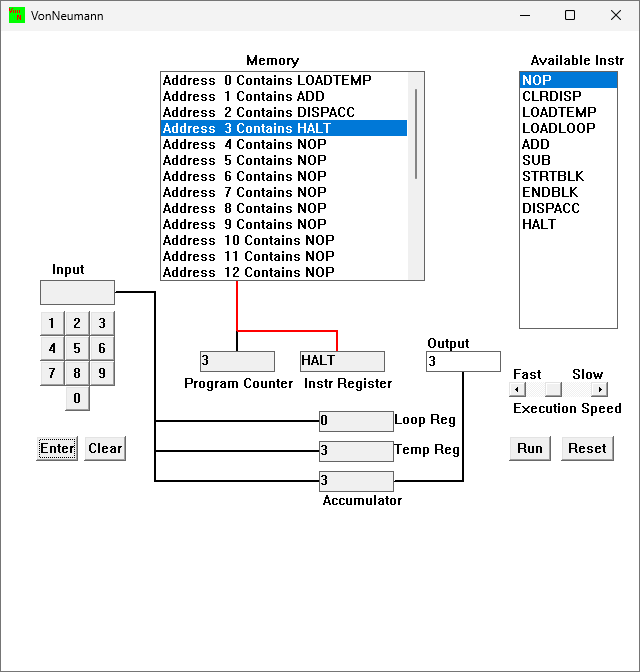
This simulator understands only the following ten instructions:

|  |  |
| --- | --- |
| 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 |

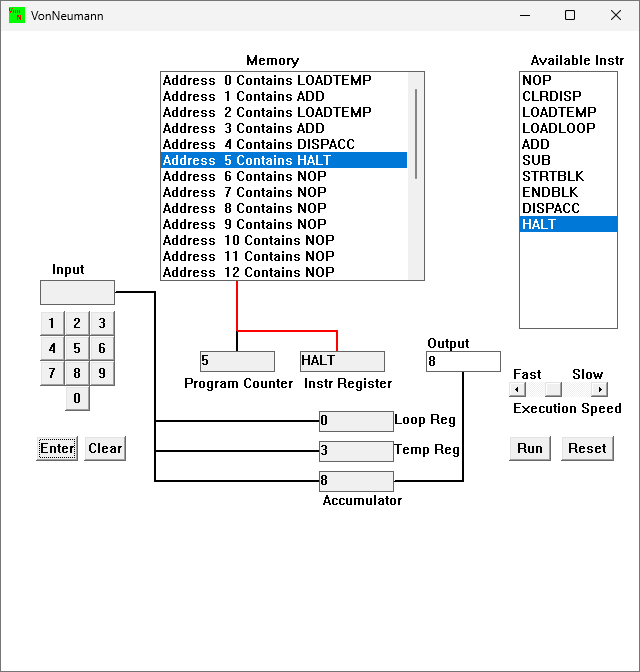
* 1. Load the following program into the memory:  
     LOADTEMP  
     ADD  
     DISPACC  
     HALT  
       
     To do this, first click on the “LOADTEMP” in the list of instructions on the right of simulator window. Then click on Memory location with “Address 0 Contains NOP”. This will then change into “Address 0 Contains LOADTEMP”. Repeat the process with “Address 1” and so on until the whole program is loaded.



* 1. Run the program by clicking on the “Run” button. The simulator would highlight the Address 0 location and then pause. It is executing the instruction “LOADTEMP” which requires you to input a number into the keypad.   
       
     Click 2 or 3 numbers on the keypad and then click the “Enter” button. The simulator will then resume running the program and execute the instruction “ADD”. This adds the number that you just entered, to the zero in the accumulator.   
       
     The next instruction is “DISPACC” which stands for “Display Accumulator”, and it does exactly that. After than the simulator stops running the program when it executes the instruction “HALT”.



* 1. Load the following program into the simulator:  
     LOADTEMP  
     ADD  
     LOADTEMP  
     ADD  
     DISPACC  
     HALT  
       
     What do you think it does? Write your answer below (10 marks)  
       
     **Answer:**

****

= list of codes:

LOADTEMP (5)

ADD

LOADTEMP (3)

ADD

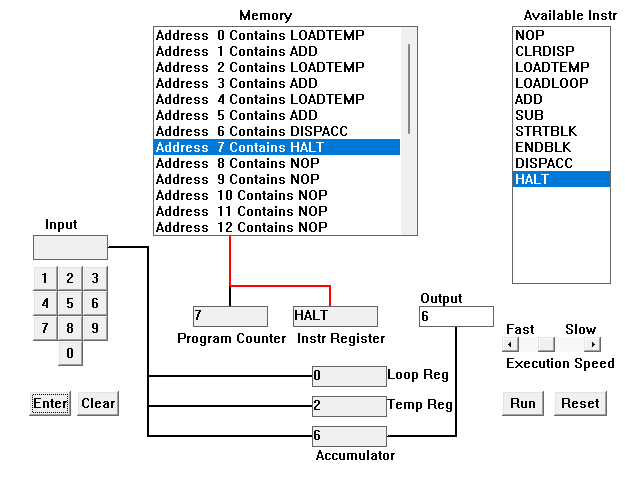
DISPACC

HALT

= In the program, it appears that the LOADTEMP instruction from memory asks for an input and loads the given input to the Temporary Register. The ADD instruction from the memory goes to the accumulator and does the addition with the next input asked by another LOADTEMP instruction. The second ADD instruction does the same work as the first ADD instruction from memory. Therefore, after both the inputs are added, the DISPACC displays the accumulator as an output and the HALT instruction stops the program from running after the work is done.

* 1. Write a program to add 3 numbers together. List your program below (10 marks)

**Answer:**



= List of codes:

LOADTEMP (2)

ADD

LOADTEMP (2)

ADD

LOADTEMP (2)

ADD

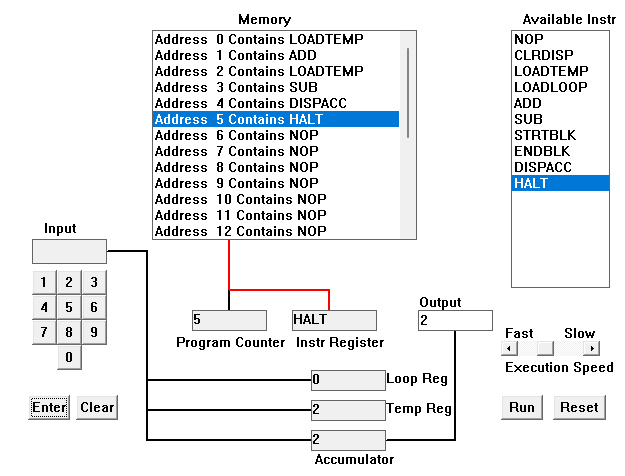
DISPACC

HALT

= In the given Program, to add the three numbers we have used three similar instructions each for ‘LOADTEMP’ and ‘ADD’. The first LOADTEMP instruction from the memory asks for an input to be entered where after the input is loaded to the temporary register. Therefore, the ADD instruction from the memory adds the first input with the second input fetched by the second LOADTEMP and the next ADD instructions again repeats the same function for the third LOADTEMP and ADD instruction in the accumulator. Then, the DISPACC displays the result of the accumulator as an output and the HALT instruction stops the program from running after an output is displayed.

* 1. Write a program to subtract a number from another. List your program below (10 marks)

**Answer:**



= list of codes:

LOADTEMP (4)

ADD

LOADTEMP (2)

SUB

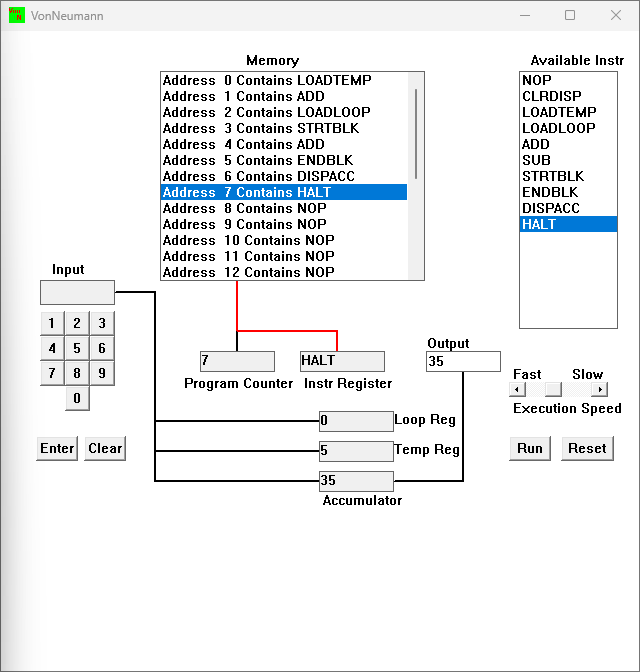
DISPACC

HALT

= In the program, it appears that the first LOADTEMP instructions asks for the input where then the input is stored in the temporary register, the ADD instruction from memory adds the data stored in the temporary register to the accumulator. The second LOADTEMP asks for another input which again gets fetched to the temporary register. The SUB instruction then performs the subtraction or subtracts the second input from the first input from the accumulator. The DISPACC instruction then displays the result from the accumulator as an output. The HALT instruction stops the program from running after the output is given.

* 1. Load the following program into the simulator:  
     LOADTEMP  
     ADD  
     LOADLOOP  
     STRTBLK  
     ADD  
     DISPACC  
     ENDBLK  
     HALT  
       
     Run it and when it reach the LOADTEMP instruction, enter 5 on the keypad and click the “Enter” button. When it reaches the LOADLOOP instruction, enter 6. What do you think the program does? Write your answer below in the form of an equation (10 marks)

**Answer:**



= list of codes:

LOADTEMP (5)

ADD

LOADLOOP (6)

STRTBLK

ADD

ENDBLK

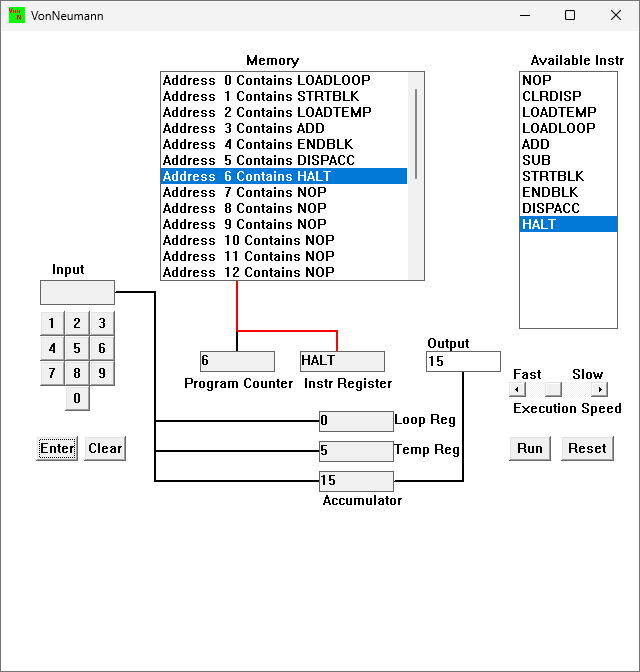
DISPACC

HALT

= in the program, we can see that the LOADTEMP instruction from memory gets an input from the user, stores it to the temporary register. Then, the ADD instruction perform addition 6 times as entered for the LOADLOOP, it adds the number 5 which is entered in the LOADTEMP, six times. The STRTBLK starts the loop so the addition can be complete and after the completion the ENDBLK instruction end the loop. Therefore, the DISPACC instruction displays the number stored in the accumulator as an output. And, when the output is given the HALT instruction stops the program from running.

* 1. Write a program that will let you add 5, or 10 or 20 numbers together. List your program below and explain how it works (25 marks)

**Answer:**



= list of codes:

LOADLOOP (5) “you can choose 5, 10, or 20 as an input as given in the question.”

STRTBLK

LOADTEMP (1, 2, 3, 4, 5)

ADD

ENDBLK

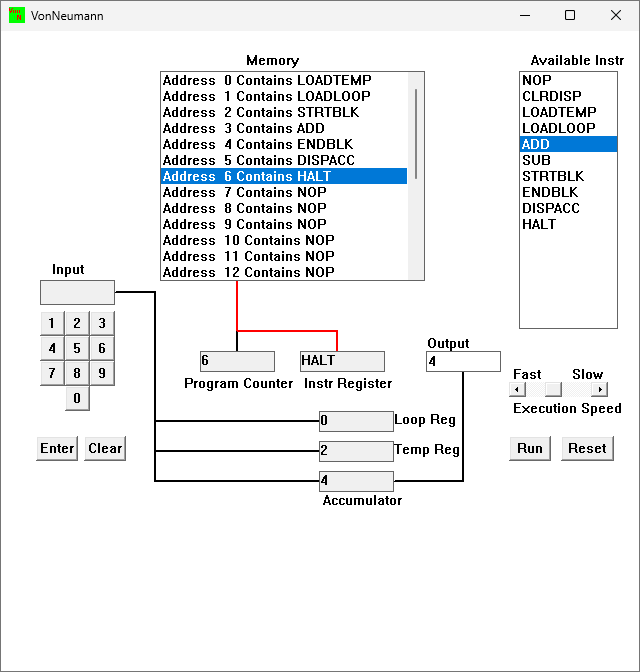
DISPACC

HALT

= In the program, we give one number as the input for LOADLOOP which means that we can do or perform addition 5 times by entering different inputs each five times in the LOADTEMP instructions. The ADD instruction performs the addition. The STRTBLK instruction instructs for the loop to start and the ENDBLK stops the loop when it comes to the final turn according to the entered input. Therefore, the DISPACC displays the result stored in the accumulator as the output and the HALT instruction stops the program from running.

* 1. Write a program that will let you multiply 2 numbers together. List your program below and explain how it works (35 marks).

**Answer:**

****

= list of code:

LOADTEMP (2)

LOADLOOP (2)

STRTBLK

ADD

ENDBLK

DISPACC

HALT

= In the program we give one number as input to the LOADTEMP which is the value to be multiplied. Then, we add an instruction LOADLOOP to the memory which takes the input from the user as to how many times they want the loop to run. Then the STARTBLK instruction starts the loop and the ADD instruction adds the LOADTEMP value in loops as we had entered in LOADLOOP. For example, in this program I first entered the value 2 in LOADTEMP to be multiplied and entered the value 2 in LOADLOOP to run the loop 2 times, and with ADD we add both values i.e. 2+2=4. And the DISPACC instruction displays the output from the accumulator and the HALT instruction stops the program from running.