CSCE 3301 – Computer Architecture

Project 2: Tomasulo Algorithm Simulation

Nour Kasaby - 900211955
Nadine Safwat - 900212508

1. Implementation

- Global Variables:
    - Multiple global variables were used to store instructions, stations, and, most importantly, flags regarding the branch, call, and ret instructions. We also used flags to see if we could write instructions at a given cycle or not.

- Reservation Stations:
    - We created a dictionary (map) for our available reservation stations. Every reservation station name points to the following fields: Name, Busy, OP, Vj, Vk, Qj, Qk, A, and Imm.
    - The reservation station fields are updated the moment the instruction is issued. As in, Y is written into Busy, and the source registers are written into Vj and Vk if they were free. Otherwise, the unit holding the source registers up will be written in Qj or Qk depending on which source register is currently being used or both Qj and Qk if both registers were not free in the current clock cycle. 'A' will have the offset stored in it up until the execution of the instruction. Once an instruction with a return address is executed, the return address will be computed and added to 'A'.
    - Later on in the program, the reservation stations are cleared once the instruction that was being used in it is written.
- Register Stat Table:
    - The register stat table indicating the business of a register is created using a dictionary that maps every register to its corresponding unit.
    - When an instruction is issued, we assign every instruction's rd (if it had a destination register) to the unit where it is currently busy until we write the instruction.
    - Once we write the instruction, the destination register will no longer be busy. Thus, we will remove the "unit" indicating that the register is now free to be used for other instructions.
- Issuing:
    - We first check if the instructions can be issued by checking if there are free reservation stations for the function.
    - If the instructions can be issued, then we modify the reservation station and flag it as busy. We also do another check to see if any of the source registers of the instructions is currently being used by another instruction; if yes, then we set Qj or Qk (or both) to the station name that is currently holding up the register, otherwise, we write the registers into Vj and/or Vk.

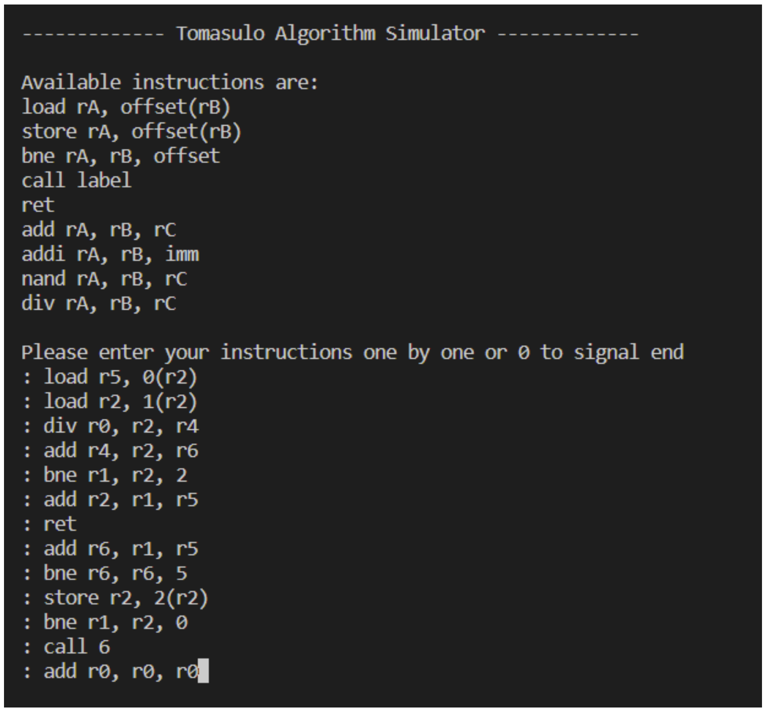
    - We set branch_issued and call_return_issued flags to true once we issued and use them later on in the program.
    - Also, when we issue instructions, we append our list of active stations with the current station that received an instruction, we append the list of issued instructions, and we append 0 in both the can_write list and the exec_time list.
- Execution:
    - We begin by checking if the instruction can be executed by seeing if its source registers are not busy.
    - The return address for instructions that have a return address is calculated during the execution of the instruction and written in ‘A’ of the reservation station.
    - Once we have finished the n cycles that take every instruction to execute, we write 1 in the can_write list using the same index as the instruction’s issued index, signaling that the current instruction has finished execution and is ready to be written.
    - We also set the start execution cycle and end execution cycle in the teaching table dictionary now to be able to print it correctly later on in the program.
- Writing:
    - We carry out the instruction’s operation in the writing stage. As in, if the instruction was a store word, then we would save the value from the chosen memory address to the destination register…etc.
    - We set the branch taken flag to true so that we can use it in later stages
    - We also fixed the reservation stations and the register stat table by freeing up the busy stations and registers so that they could be used for other instructions.
- Simulate:
    - We assign the stall issuing and stall executing flags in order to account for the fact that the program should stop issuing further instructions once it has issued a return or a call until it writes them back, and it should stop executing instructions once it has issued a branch until it writes it back.
    - Based on clock cycle conditions, we carry out the appropriate step whether to issue, execute, or write. When we issue, we add the instructions along with the clock cycle to the trace table dictionary to be printed later on. We also will check if a branch was taken or if either a call or a ret instruction was written. If yes, then we set the stalling flags to false, and we removed the reservation station and the instructions issued but were skipped due to the branch being taken.

    - We also added the writing clock cycle in the tracing table and popped the instructions that were in the write_queue as they have been written and are no longer waiting to be written.
- Output Format:
    - We used the ‘tabulate’ library in Python to output the results as tables. Our output has the tracing table, the reservation station table, the register file table, and the register stat table. Then, after you signal the end of the program by entering a 0, the metrics of the program will show up, which consist of the branch misprediction percentage, the IPC, and the clock cycles taken.
- Main:
    - We finally combine everything together in a main function called top().
- **Note:**
    - We have not explicitly done a bonus; however, we do output all our tables **per clock cycle** so one can see what happens in every instruction by stepping through the program cycle by cycle. This satisfies the bonus criteria mentioned in the educational GUI part, except that we output the results from the console rather than implementing a GUI.

2. User Guide

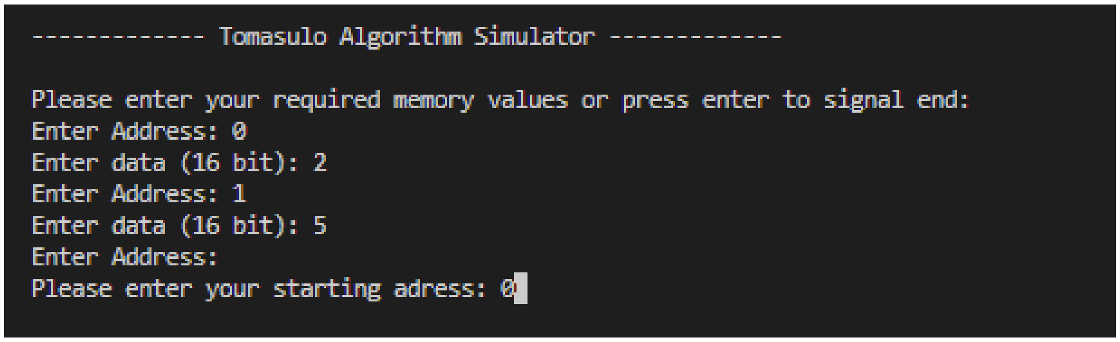
Inputs:

To enter the instructions for your program all you have to do is type them out one by one, and the simulator will check that you have entered the correct instruction. To signal that you have written all your instructions you must enter 0.



```
------------- Tomasulo Algorithm Simulator -------------

Available instructions are:
load rA, offset(rB)
store rA, offset(rB)
bne rA, rB, offset
call label
ret
add rA, rB, rC
addi rA, rB, imm
nand rA, rB, rC
div rA, rB, rC

Please enter your instructions one by one or 0 to signal end
: load r5, 0(r2)
: load r2, 1(r2)
: div r0, r2, r4
: add r4, r2, r6
: bne r1, r2, 2
: add r2, r1, r5
: ret
: add r6, r1, r5
: bne r6, r6, 5
: store r2, 2(r2)
: bne r1, r2, 0
: call 6
: add r0, r0, r0
```

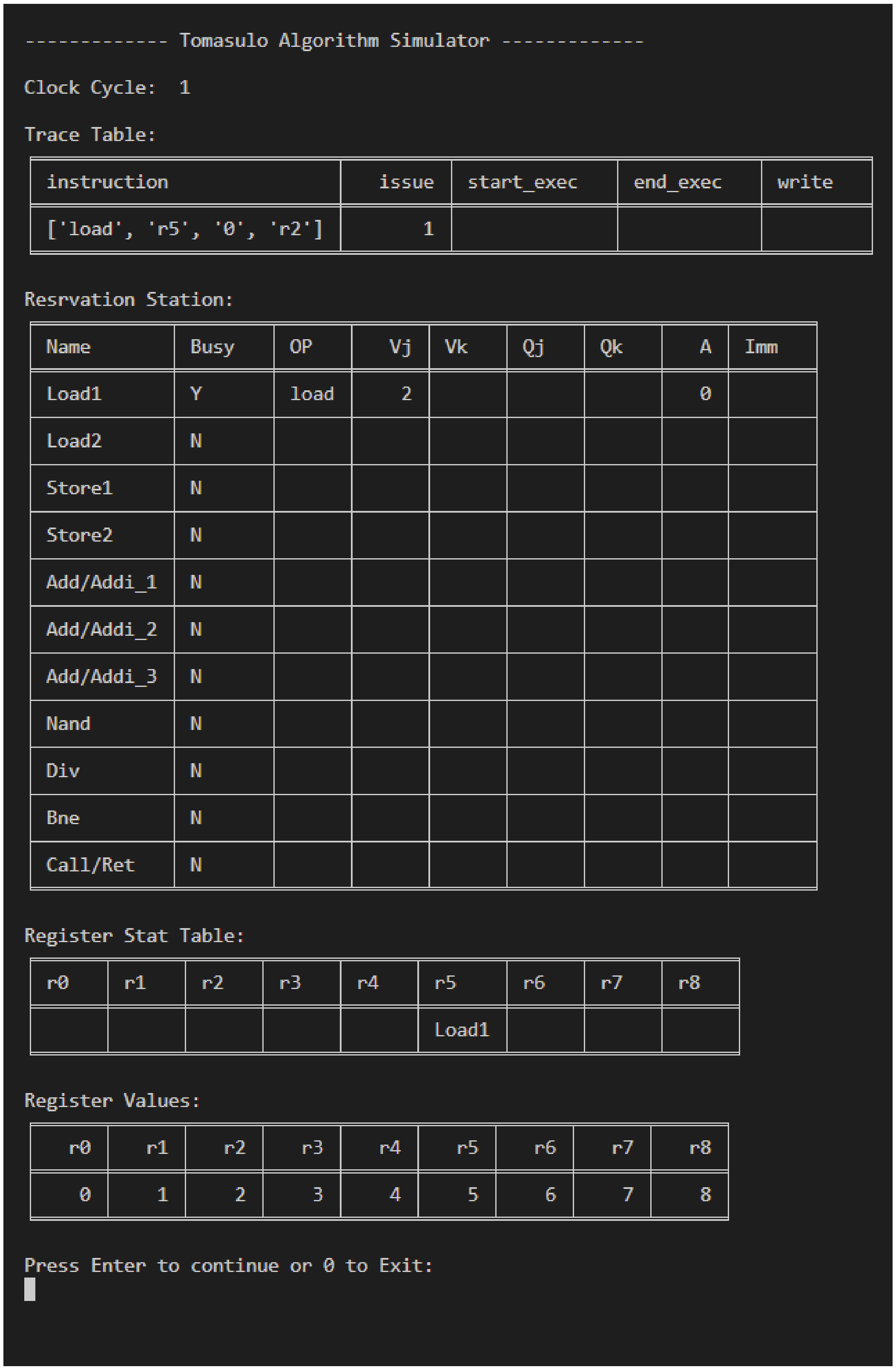
To enter the data required into memory all you have to do is input the address and the data you would like to enter. You must press enter to signal you have finished entering data. Then you will be asked to enter a starting address and after that you will be automatically sent to the simulator.



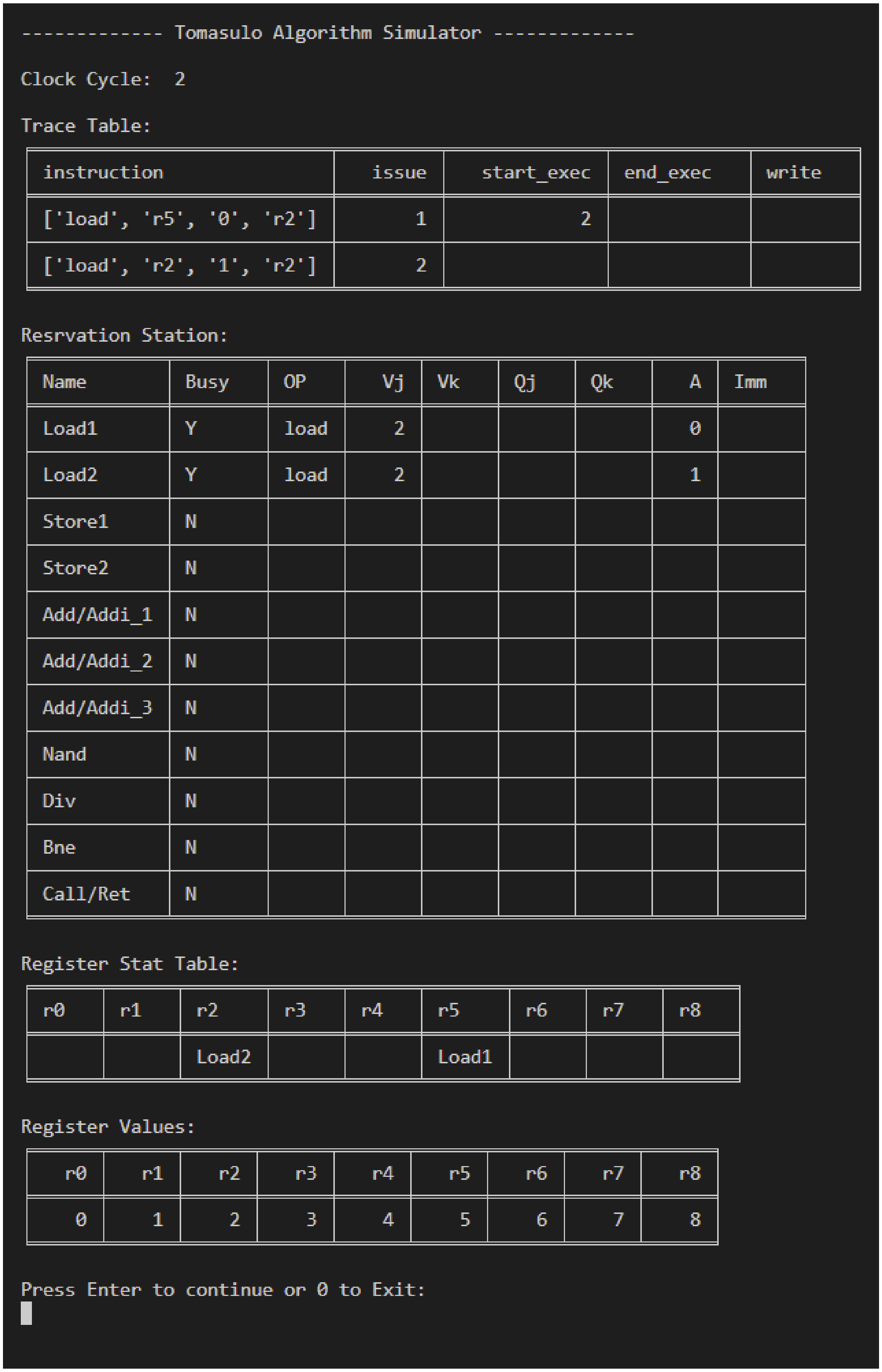
```
------------- Tomasulo Algorithm Simulator -------------

Please enter your required memory values or press enter to signal end:
Enter Address: 0
Enter data (16 bit): 2
Enter Address: 1
Enter data (16 bit): 5
Enter Address:
Please enter your starting adress: 0
```

Cycles:

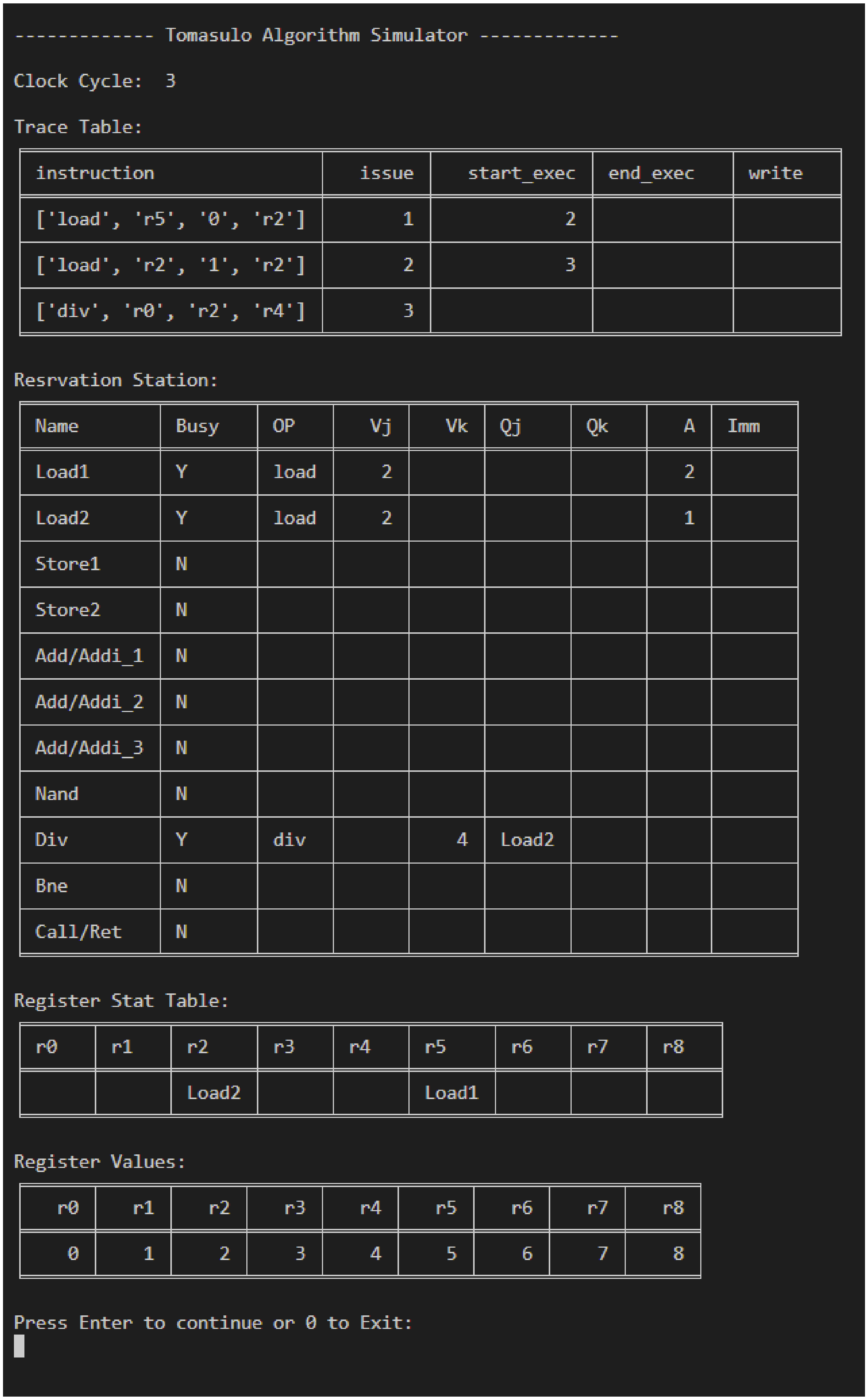
To move on to the next cycle you must press enter. You can also press 0 to leave the simulation.

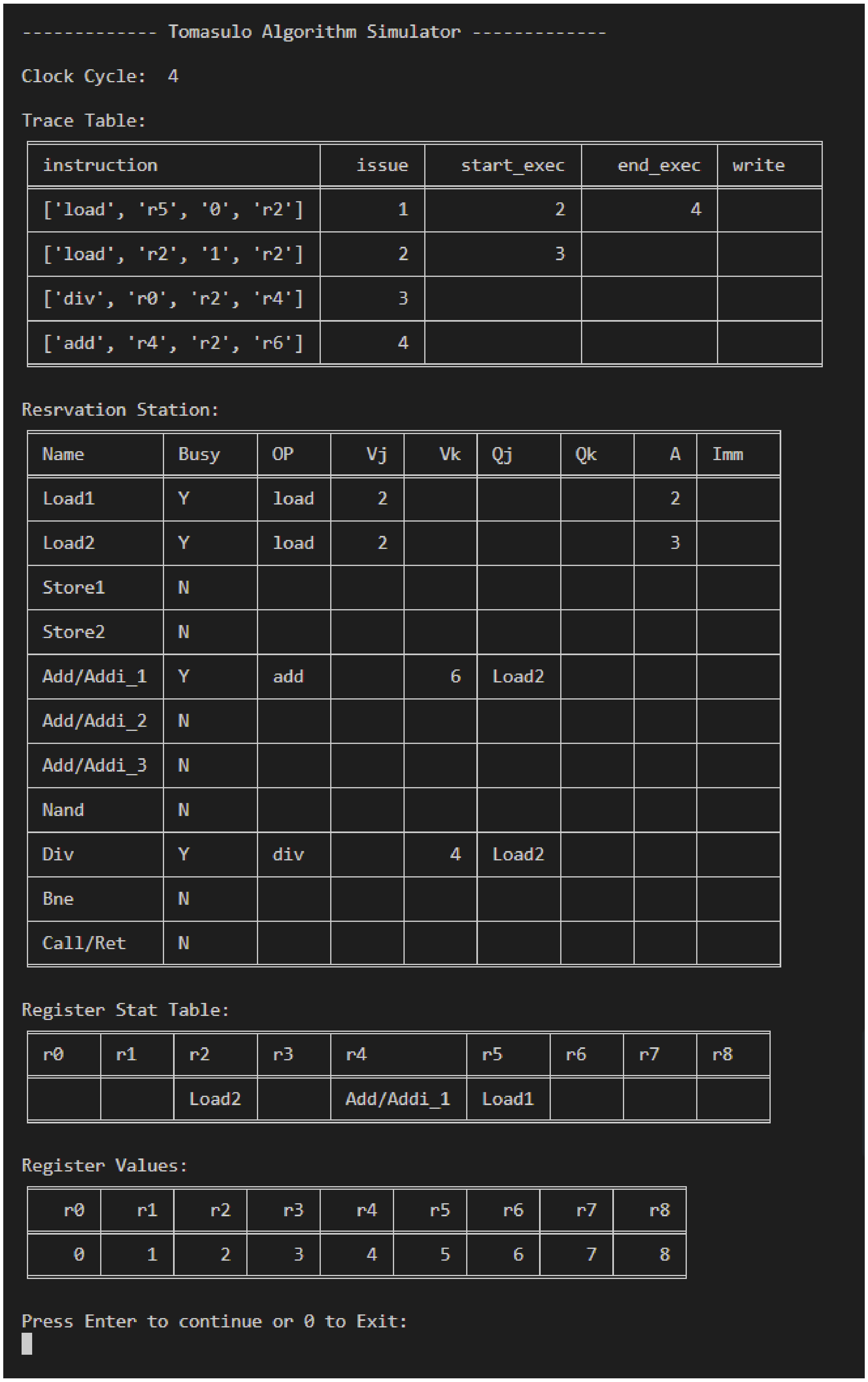


```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  1

Trace Table:
+-----------------------------+-------+------------+----------+-------+
| instruction                 | issue | start_exec | end_exec | write |
+-----------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']   |     1 |            |          |       |
+-----------------------------+-------+------------+----------+-------+

Resrvation Station:
+-------------+------+------+----+----+----+----+---+-----+
| Name        | Busy | OP   | Vj | Vk | Qj | Qk | A | Imm |
+-------------+------+------+----+----+----+----+---+-----+
| Load1       | Y    | load |  2 |    |    |    | 0 |     |
| Load2       | N    |      |    |    |    |    |   |     |
| Store1      | N    |      |    |    |    |    |   |     |
| Store2      | N    |      |    |    |    |    |   |     |
| Add/Addi_1  | N    |      |    |    |    |    |   |     |
| Add/Addi_2  | N    |      |    |    |    |    |   |     |
| Add/Addi_3  | N    |      |    |    |    |    |   |     |
| Nand        | N    |      |    |    |    |    |   |     |
| Div         | N    |      |    |    |    |    |   |     |
| Bne         | N    |      |    |    |    |    |   |     |
| Call/Ret    | N    |      |    |    |    |    |   |     |
+-------------+------+------+----+----+----+----+---+-----+

Register Stat Table:
+----+----+----+----+----+-------+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5    | r6 | r7 | r8 |
+----+----+----+----+----+-------+----+----+----+
|    |    |    |    |    | Load1 |    |    |    |
+----+----+----+----+----+-------+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 |  1 |  2 |  3 |  4 |  5 |  6 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  2

Trace Table:
╒══════════════════════════════╤═════════╤══════════════╤════════════╤═════════╕
│ instruction                  │   issue │   start_exec │ end_exec   │ write   │
╞══════════════════════════════╪═════════╪══════════════╪════════════╪═════════╡
│ ['load', 'r5', '0', 'r2']    │       1 │            2 │            │         │
├──────────────────────────────┼─────────┼──────────────┼────────────┼─────────┤
│ ['load', 'r2', '1', 'r2']    │       2 │              │            │         │
╘══════════════════════════════╧═════════╧══════════════╧════════════╧═════════╛

Resrvation Station:
╒═════════════╤════════╤══════╤══════╤══════╤══════╤══════╤═════╤═══════╕
│ Name        │ Busy   │ OP   │   Vj │ Vk   │ Qj   │ Qk   │   A │ Imm   │
╞═════════════╪════════╪══════╪══════╪══════╪══════╪══════╪═════╪═══════╡
│ Load1       │ Y      │ load │    2 │      │      │      │   0 │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Load2       │ Y      │ load │    2 │      │      │      │   1 │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Store1      │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Store2      │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Add/Addi_1  │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Add/Addi_2  │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Add/Addi_3  │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Nand        │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Div         │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Bne         │ N      │      │      │      │      │      │     │       │
├─────────────┼────────┼──────┼──────┼──────┼──────┼──────┼─────┼───────┤
│ Call/Ret    │ N      │      │      │      │      │      │     │       │
╘═════════════╧════════╧══════╧══════╧══════╧══════╧══════╧═════╧═══════╛

Register Stat Table:
╒══════╤══════╤═══════╤══════╤══════╤═══════╤══════╤══════╤══════╕
│ r0   │ r1   │ r2    │ r3   │ r4   │ r5    │ r6   │ r7   │ r8   │
╞══════╪══════╪═══════╪══════╪══════╪═══════╪══════╪══════╪══════╡
│      │      │ Load2 │      │      │ Load1 │      │      │      │
╘══════╧══════╧═══════╧══════╧══════╧═══════╧══════╧══════╧══════╛

Register Values:
╒══════╤══════╤══════╤══════╤══════╤══════╤══════╤══════╤══════╕
│   r0 │   r1 │   r2 │   r3 │   r4 │   r5 │   r6 │   r7 │   r8 │
╞══════╪══════╪══════╪══════╪══════╪══════╪══════╪══════╪══════╡
│    0 │    1 │    2 │    3 │    4 │    5 │    6 │    7 │    8 │
╘══════╧══════╧══════╧══════╧══════╧══════╧══════╧══════╧══════╛

Press Enter to continue or 0 to Exit:
█
```


```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  3

Trace Table:
+--------------------------------+---------+--------------+------------+---------+
| instruction                    |   issue |   start_exec | end_exec   | write   |
+================================+=========+==============+============+=========+
| ['load', 'r5', '0', 'r2']      |       1 |            2 |            |         |
+--------------------------------+---------+--------------+------------+---------+
| ['load', 'r2', '1', 'r2']      |       2 |            3 |            |         |
+--------------------------------+---------+--------------+------------+---------+
| ['div', 'r0', 'r2', 'r4']      |       3 |              |            |         |
+--------------------------------+---------+--------------+------------+---------+

Resrvation Station:
+-------------+--------+------+------+------+-------+------+-----+-------+
| Name        | Busy   | OP   |   Vj |   Vk | Qj    | Qk   |   A | Imm   |
+=============+========+======+======+======+=======+======+=====+=======+
| Load1       | Y      | load |    2 |      |       |      |   2 |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Load2       | Y      | load |    2 |      |       |      |   1 |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Store1      | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Store2      | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Add/Addi_1  | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Add/Addi_2  | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Add/Addi_3  | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Nand        | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Div         | Y      | div  |      |    4 | Load2 |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Bne         | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+
| Call/Ret    | N      |      |      |      |       |      |     |       |
+-------------+--------+------+------+------+-------+------+-----+-------+

Register Stat Table:
+------+------+-------+------+------+-------+------+------+------+
| r0   | r1   | r2    | r3   | r4   | r5    | r6   | r7   | r8   |
+======+======+=======+======+======+=======+======+======+======+
|      |      | Load2 |      |      | Load1 |      |      |      |
+------+------+-------+------+------+-------+------+------+------+

Register Values:
+------+------+------+------+------+------+------+------+------+
|   r0 |   r1 |   r2 |   r3 |   r4 |   r5 |   r6 |   r7 |   r8 |
+======+======+======+======+======+======+======+======+======+
|    0 |    1 |    2 |    3 |    4 |    5 |    6 |    7 |    8 |
+------+------+------+------+------+------+------+------+------+

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  4

Trace Table:
```

instruction	issue	start_exec	end_exec	write
['load', 'r5', '0', 'r2']	1	2	4	
['load', 'r2', '1', 'r2']	2	3		
['div', 'r0', 'r2', 'r4']	3			
['add', 'r4', 'r2', 'r6']	4			

```
Resrvation Station:
```

Name	Busy	OP	Vj	Vk	Qj	Qk	A	Imm
Load1	Y	load	2				2	
Load2	Y	load	2				3	
Store1	N							
Store2	N							
Add/Addi_1	Y	add		6	Load2			
Add/Addi_2	N							
Add/Addi_3	N							
Nand	N							
Div	Y	div		4	Load2			
Bne	N							
Call/Ret	N							

```
Register Stat Table:
```

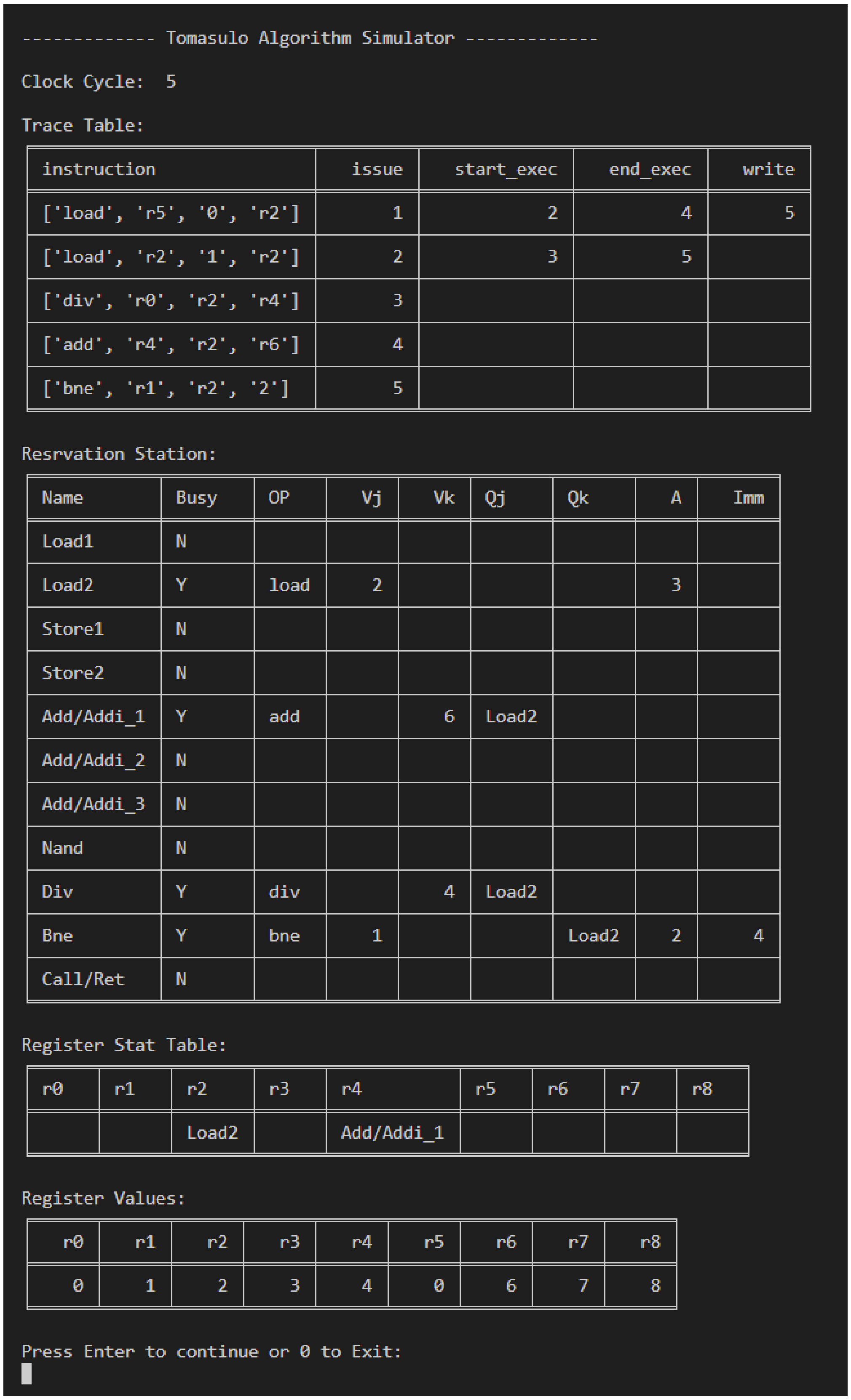
r0	r1	r2	r3	r4	r5	r6	r7	r8
		Load2		Add/Addi_1	Load1			

```
Register Values:
```

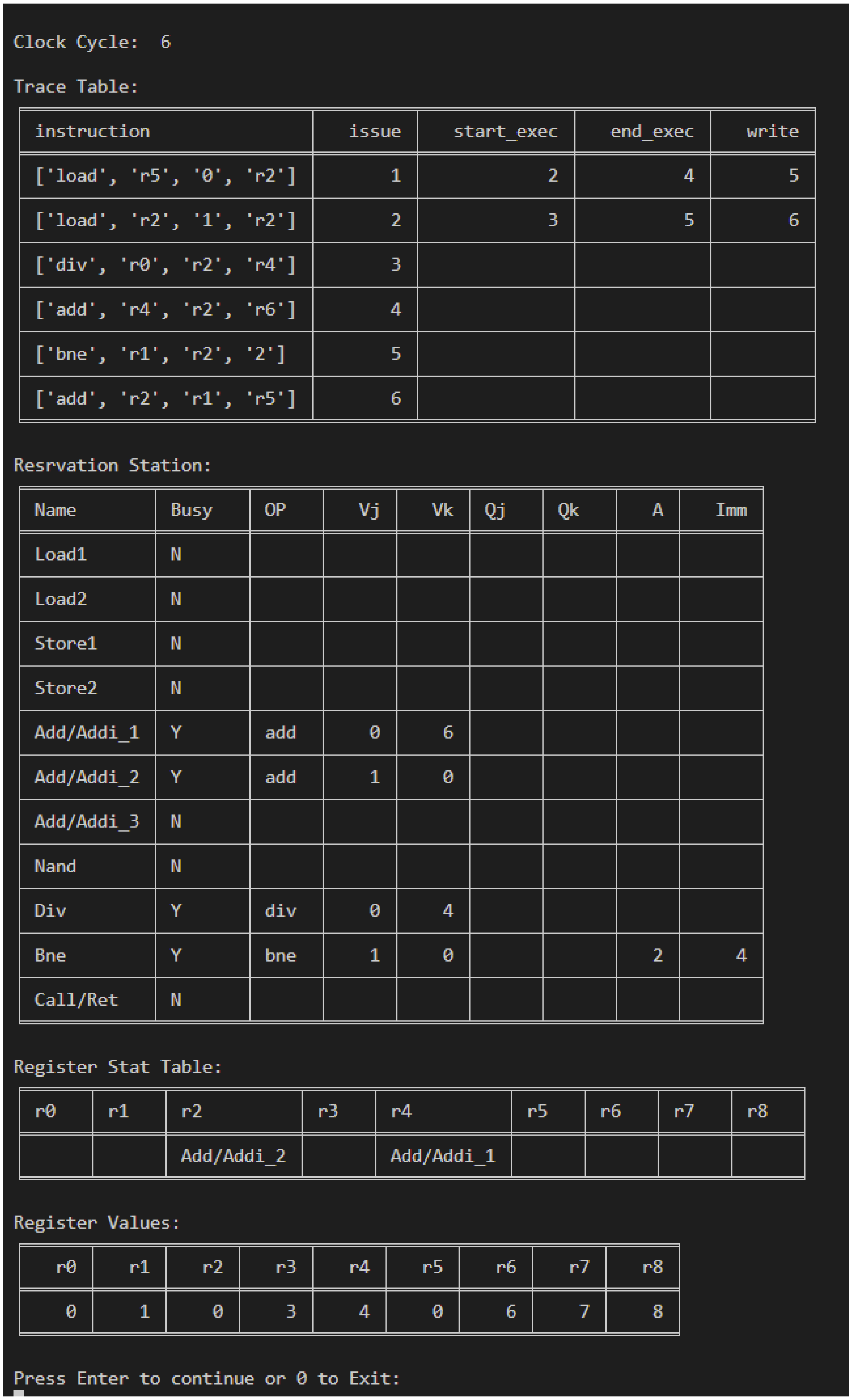
r0	r1	r2	r3	r4	r5	r6	r7	r8
0	1	2	3	4	5	6	7	8

```
Press Enter to continue or 0 to Exit:
```

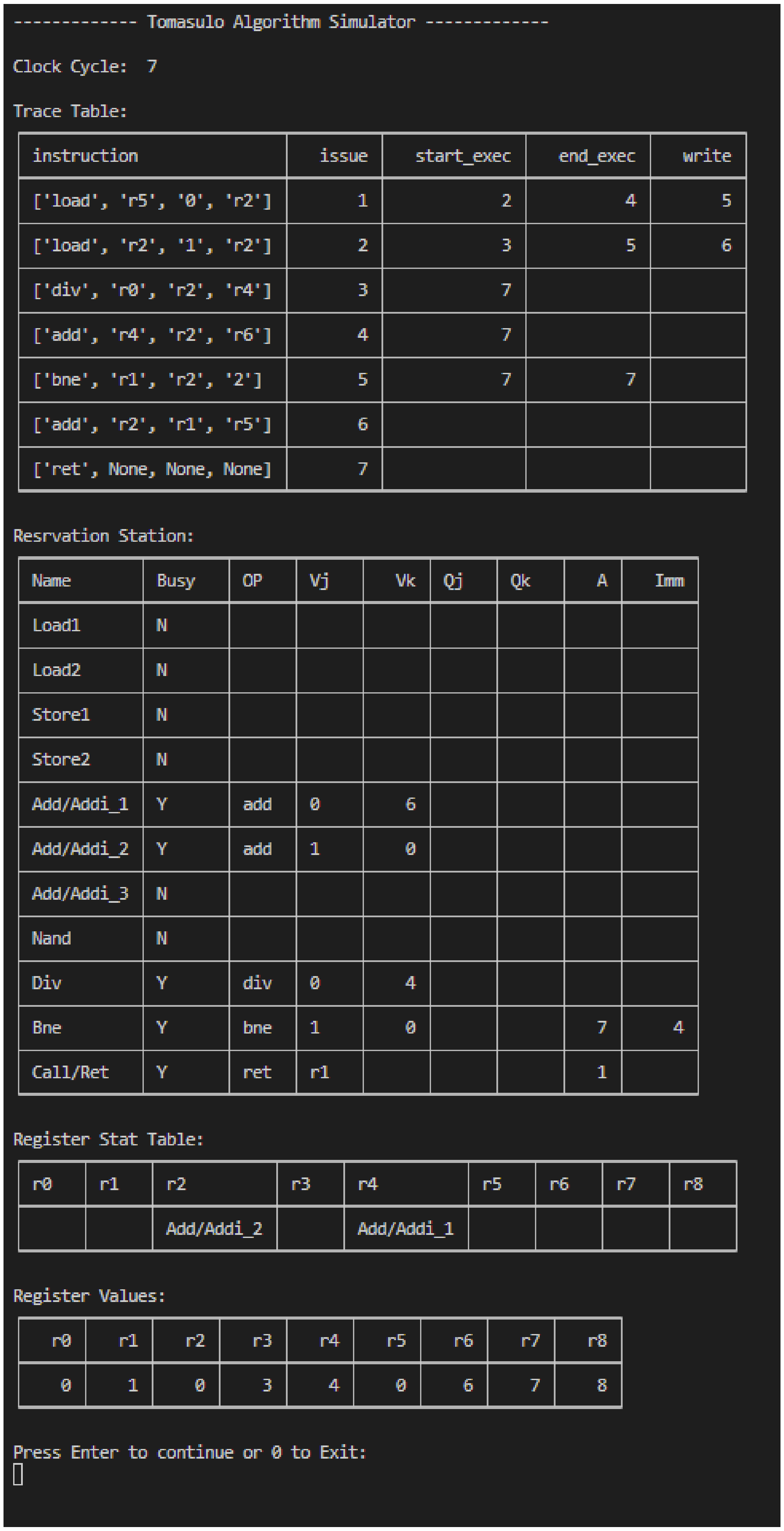
In the following cycles you will see that div add and bne will not start exec until the second load is written



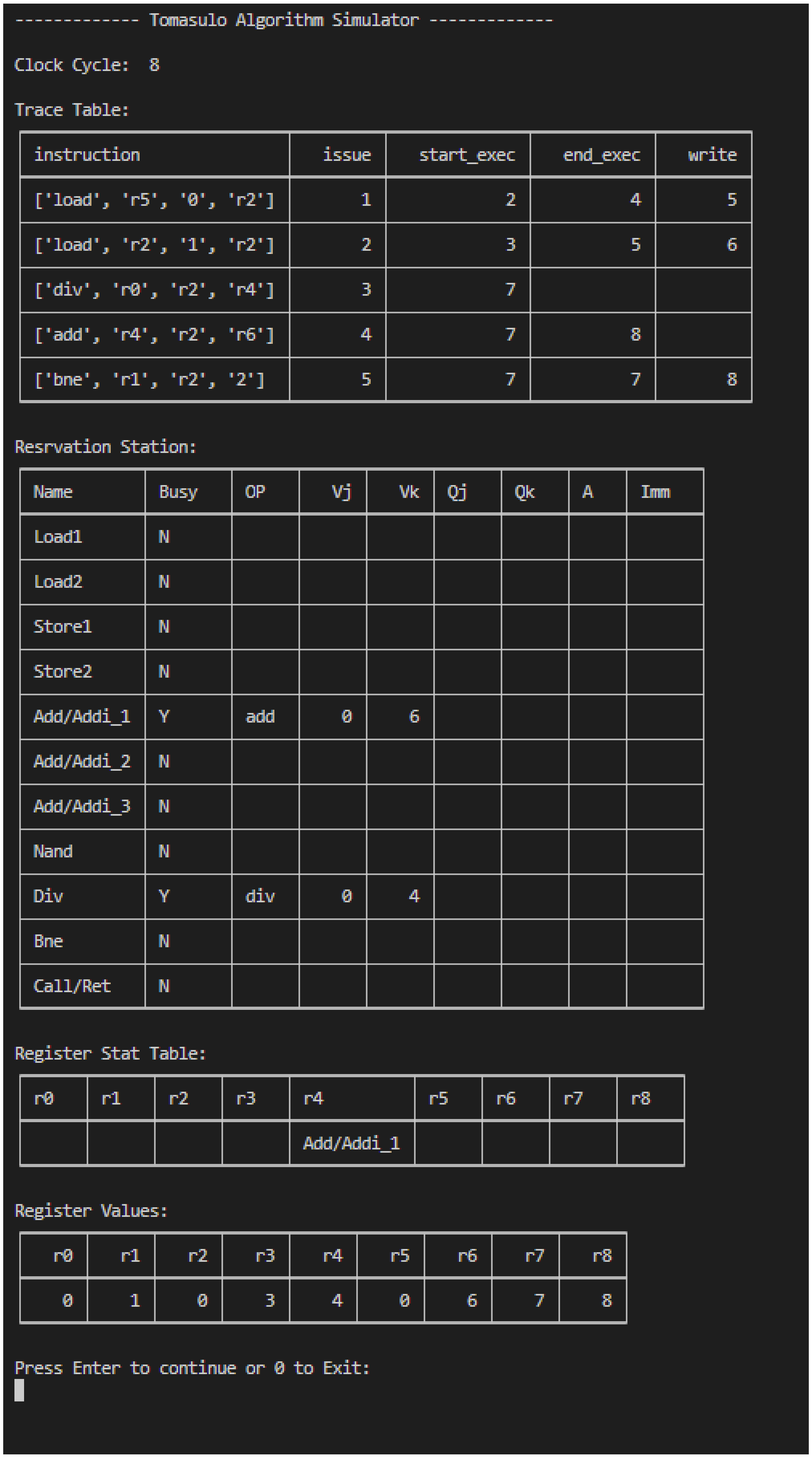
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  5

Trace Table:
| instruction                  |   issue |   start_exec |   end_exec |   write |
|------------------------------|---------|--------------|------------|---------|
| ['load', 'r5', '0', 'r2']    |       1 |            2 |          4 |       5 |
| ['load', 'r2', '1', 'r2']    |       2 |            3 |          5 |         |
| ['div', 'r0', 'r2', 'r4']    |       3 |              |            |         |
| ['add', 'r4', 'r2', 'r6']    |       4 |              |            |         |
| ['bne', 'r1', 'r2', '2']     |       5 |              |            |         |

Resrvation Station:
| Name        | Busy   | OP    |   Vj |   Vk | Qj     | Qk     |   A |   Imm |
|-------------|--------|-------|------|------|--------|--------|-----|-------|
| Load1       | N      |       |      |      |        |        |     |       |
| Load2       | Y      | load  |    2 |      |        |        |   3 |       |
| Store1      | N      |       |      |      |        |        |     |       |
| Store2      | N      |       |      |      |        |        |     |       |
| Add/Addi_1  | Y      | add   |      |    6 | Load2  |        |     |       |
| Add/Addi_2  | N      |       |      |      |        |        |     |       |
| Add/Addi_3  | N      |       |      |      |        |        |     |       |
| Nand        | N      |       |      |      |        |        |     |       |
| Div         | Y      | div   |      |    4 | Load2  |        |     |       |
| Bne         | Y      | bne   |    1 |      |        | Load2  |   2 |     4 |
| Call/Ret    | N      |       |      |      |        |        |     |       |

Register Stat Table:
| r0   | r1   | r2    | r3   | r4          | r5   | r6   | r7   | r8   |
|------|------|-------|------|-------------|------|------|------|------|
|      |      | Load2 |      | Add/Addi_1  |      |      |      |      |

Register Values:
|   r0 |   r1 |   r2 |   r3 |   r4 |   r5 |   r6 |   r7 |   r8 |
|------|------|------|------|------|------|------|------|------|
|    0 |    1 |    2 |    3 |    4 |    0 |    6 |    7 |    8 |

Press Enter to continue or 0 to Exit:
```



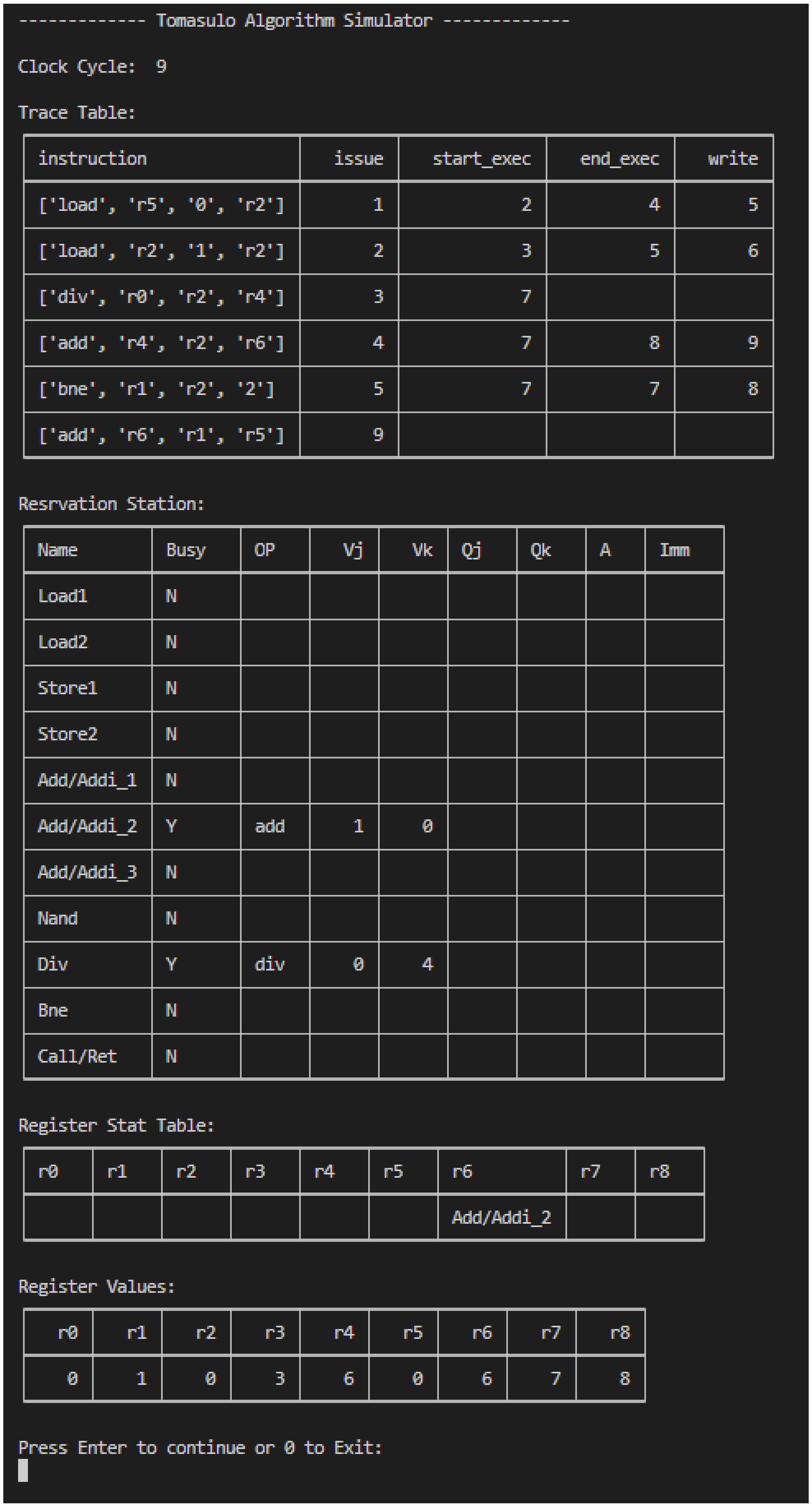
```
Clock Cycle:  6

Trace Table:
+-----------------------------+-------+------------+----------+-------+
| instruction                 | issue | start_exec | end_exec | write |
+=============================+=======+============+==========+=======+
| ['load', 'r5', '0', 'r2']   |     1 |          2 |        4 |     5 |
+-----------------------------+-------+------------+----------+-------+
| ['load', 'r2', '1', 'r2']   |     2 |          3 |        5 |     6 |
+-----------------------------+-------+------------+----------+-------+
| ['div', 'r0', 'r2', 'r4']   |     3 |            |          |       |
+-----------------------------+-------+------------+----------+-------+
| ['add', 'r4', 'r2', 'r6']   |     4 |            |          |       |
+-----------------------------+-------+------------+----------+-------+
| ['bne', 'r1', 'r2', '2']    |     5 |            |          |       |
+-----------------------------+-------+------------+----------+-------+
| ['add', 'r2', 'r1', 'r5']   |     6 |            |          |       |
+-----------------------------+-------+------------+----------+-------+

Resrvation Station:
+-------------+------+-----+----+----+----+----+---+-----+
| Name        | Busy | OP  | Vj | Vk | Qj | Qk | A | Imm |
+=============+======+=====+====+====+====+====+===+=====+
| Load1       | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Load2       | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Store1      | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Store2      | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Add/Addi_1  | Y    | add |  0 |  6 |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Add/Addi_2  | Y    | add |  1 |  0 |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Add/Addi_3  | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Nand        | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Div         | Y    | div |  0 |  4 |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+
| Bne         | Y    | bne |  1 |  0 |    |    | 2 |   4 |
+-------------+------+-----+----+----+----+----+---+-----+
| Call/Ret    | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+

Register Stat Table:
+----+----+------------+----+------------+----+----+----+----+
| r0 | r1 | r2         | r3 | r4         | r5 | r6 | r7 | r8 |
+====+====+============+====+============+====+====+====+====+
|    |    | Add/Addi_2 |    | Add/Addi_1 |    |    |    |    |
+----+----+------------+----+------------+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+====+====+====+====+====+====+====+====+====+
|  0 |  1 |  0 |  3 |  4 |  0 |  6 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  7

Trace Table:
+------------------------------+---------+--------------+------------+---------+
| instruction                  |   issue |   start_exec |   end_exec |   write |
+------------------------------+---------+--------------+------------+---------+
| ['load', 'r5', '0', 'r2']    |       1 |            2 |          4 |       5 |
| ['load', 'r2', '1', 'r2']    |       2 |            3 |          5 |       6 |
| ['div', 'r0', 'r2', 'r4']    |       3 |            7 |            |         |
| ['add', 'r4', 'r2', 'r6']    |       4 |            7 |            |         |
| ['bne', 'r1', 'r2', '2']     |       5 |            7 |          7 |         |
| ['add', 'r2', 'r1', 'r5']    |       6 |              |            |         |
| ['ret', None, None, None]    |       7 |              |            |         |
+------------------------------+---------+--------------+------------+---------+

Resrvation Station:
+-------------+--------+------+------+------+------+------+-----+-------+
| Name        | Busy   | OP   | Vj   |   Vk | Qj   | Qk   |   A |   Imm |
+-------------+--------+------+------+------+------+------+-----+-------+
| Load1       | N      |      |      |      |      |      |     |       |
| Load2       | N      |      |      |      |      |      |     |       |
| Store1      | N      |      |      |      |      |      |     |       |
| Store2      | N      |      |      |      |      |      |     |       |
| Add/Addi_1  | Y      | add  | 0    |    6 |      |      |     |       |
| Add/Addi_2  | Y      | add  | 1    |    0 |      |      |     |       |
| Add/Addi_3  | N      |      |      |      |      |      |     |       |
| Nand        | N      |      |      |      |      |      |     |       |
| Div         | Y      | div  | 0    |    4 |      |      |     |       |
| Bne         | Y      | bne  | 1    |    0 |      |      |   7 |     4 |
| Call/Ret    | Y      | ret  | r1   |      |      |      |   1 |       |
+-------------+--------+------+------+------+------+------+-----+-------+

Register Stat Table:
+------+------+------------+------+------------+------+------+------+------+
| r0   | r1   | r2         | r3   | r4         | r5   | r6   | r7   | r8   |
+------+------+------------+------+------------+------+------+------+------+
|      |      | Add/Addi_2 |      | Add/Addi_1 |      |      |      |      |
+------+------+------------+------+------------+------+------+------+------+

Register Values:
+------+------+------+------+------+------+------+------+------+
|   r0 |   r1 |   r2 |   r3 |   r4 |   r5 |   r6 |   r7 |   r8 |
+------+------+------+------+------+------+------+------+------+
|    0 |    1 |    0 |    3 |    4 |    0 |    6 |    7 |    8 |
+------+------+------+------+------+------+------+------+------+


Press Enter to continue or 0 to Exit:
▯
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  8

Trace Table:

 instruction                    issue   start_exec   end_exec   write
 ['load', 'r5', '0', 'r2']          1            2          4       5
 ['load', 'r2', '1', 'r2']          2            3          5       6
 ['div', 'r0', 'r2', 'r4']          3            7
 ['add', 'r4', 'r2', 'r6']          4            7          8
 ['bne', 'r1', 'r2', '2']           5            7          7       8

Resrvation Station:

 Name         Busy   OP      Vj    Vk   Qj   Qk   A   Imm
 Load1        N
 Load2        N
 Store1       N
 Store2       N
 Add/Addi_1   Y      add      0     6
 Add/Addi_2   N
 Add/Addi_3   N
 Nand         N
 Div          Y      div      0     4
 Bne          N
 Call/Ret     N

Register Stat Table:

 r0   r1   r2   r3   r4           r5   r6   r7   r8
                     Add/Addi_1

Register Values:

  r0   r1   r2   r3   r4   r5   r6   r7   r8
   0    1    0    3    4    0    6    7    8

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  9

Trace Table:
```

instruction	issue	start_exec	end_exec	write
['load', 'r5', '0', 'r2']	1	2	4	5
['load', 'r2', '1', 'r2']	2	3	5	6
['div', 'r0', 'r2', 'r4']	3	7		
['add', 'r4', 'r2', 'r6']	4	7	8	9
['bne', 'r1', 'r2', '2']	5	7	7	8
['add', 'r6', 'r1', 'r5']	9			

```
Resrvation Station:
```

Name	Busy	OP	Vj	Vk	Qj	Qk	A	Imm
Load1	N							
Load2	N							
Store1	N							
Store2	N							
Add/Addi_1	N							
Add/Addi_2	Y	add	1	0				
Add/Addi_3	N							
Nand	N							
Div	Y	div	0	4				
Bne	N							
Call/Ret	N							

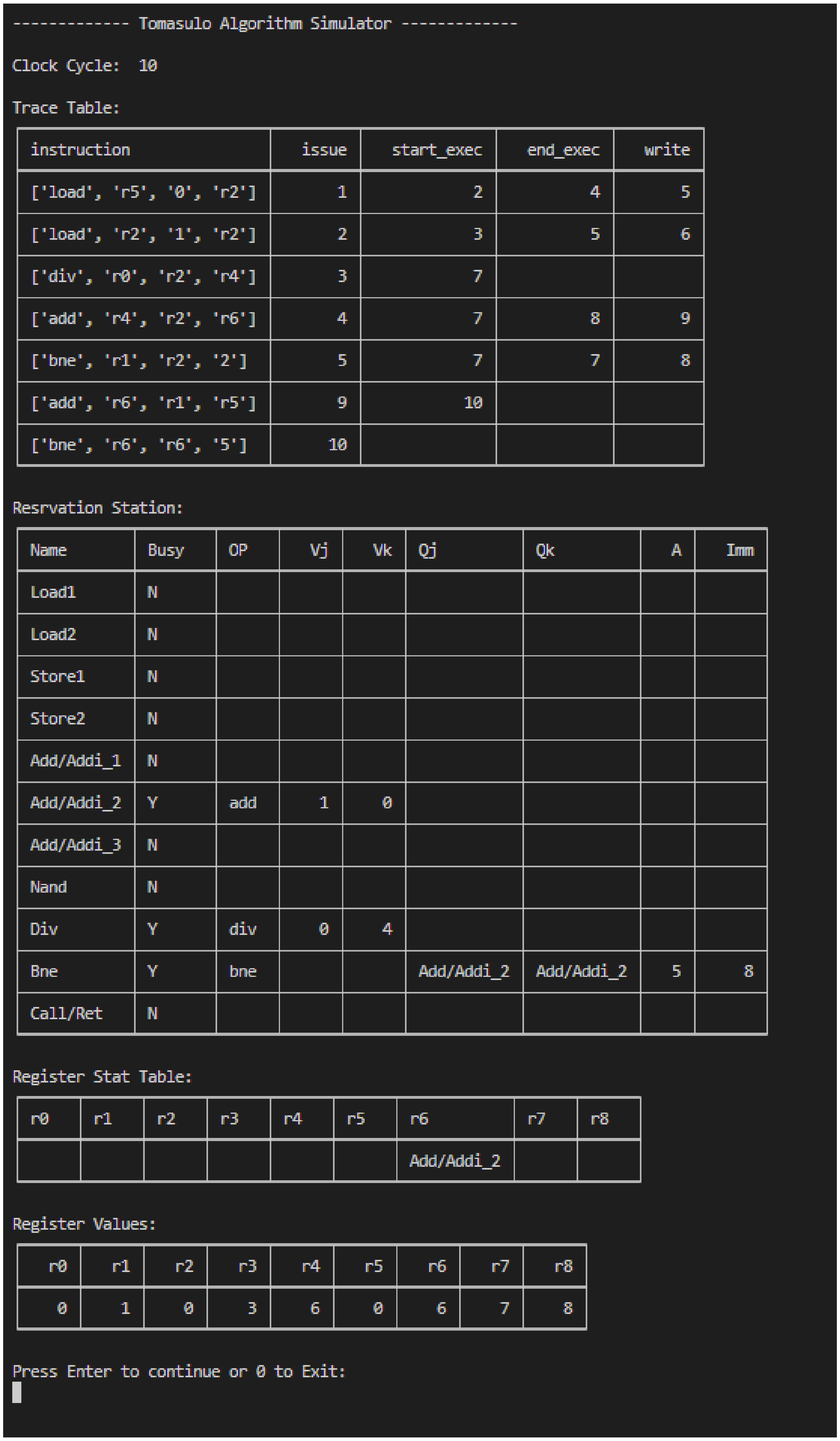
```
Register Stat Table:
```

r0	r1	r2	r3	r4	r5	r6	r7	r8
						Add/Addi_2		

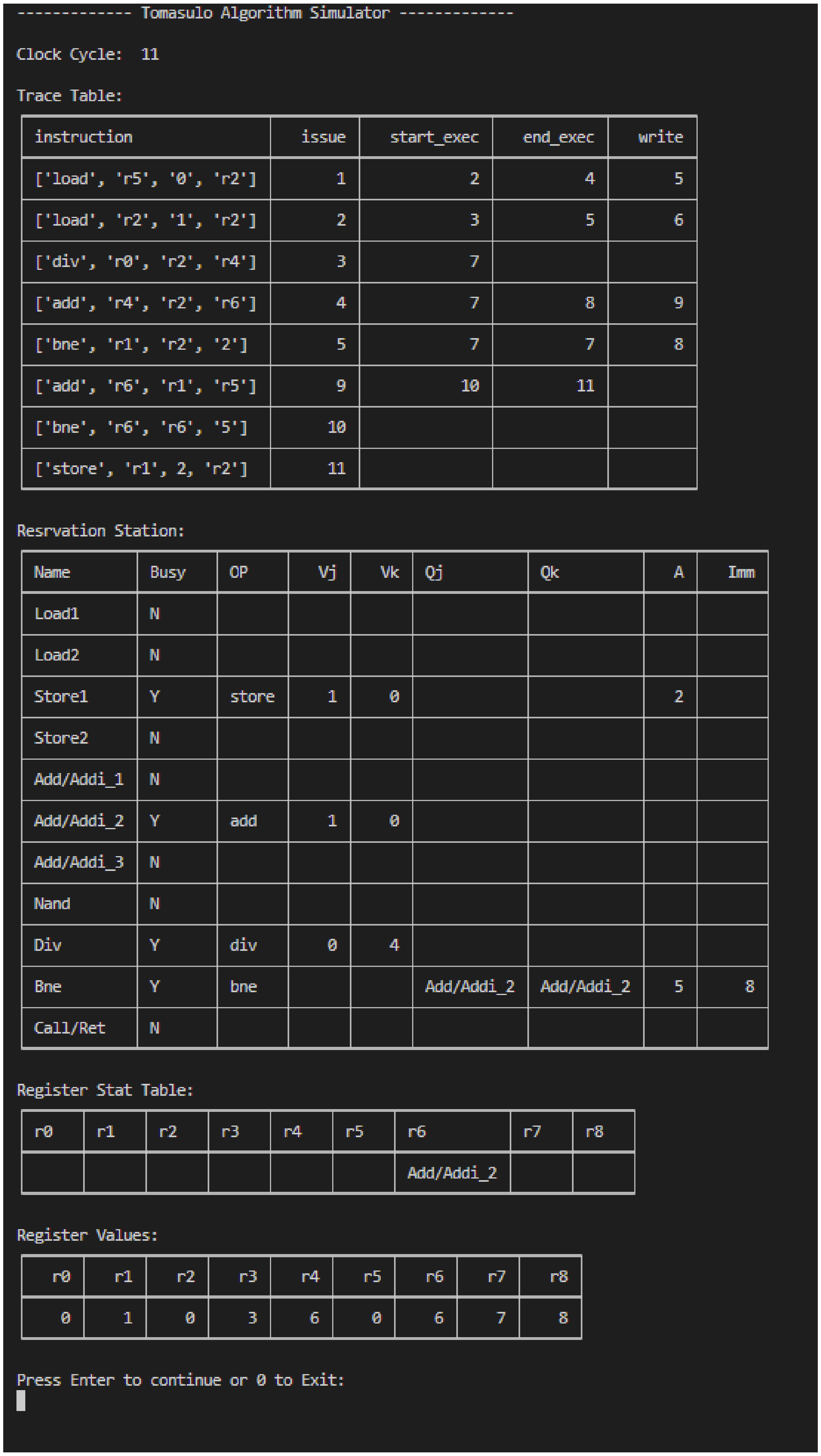
```
Register Values:
```

r0	r1	r2	r3	r4	r5	r6	r7	r8
0	1	0	3	6	0	6	7	8

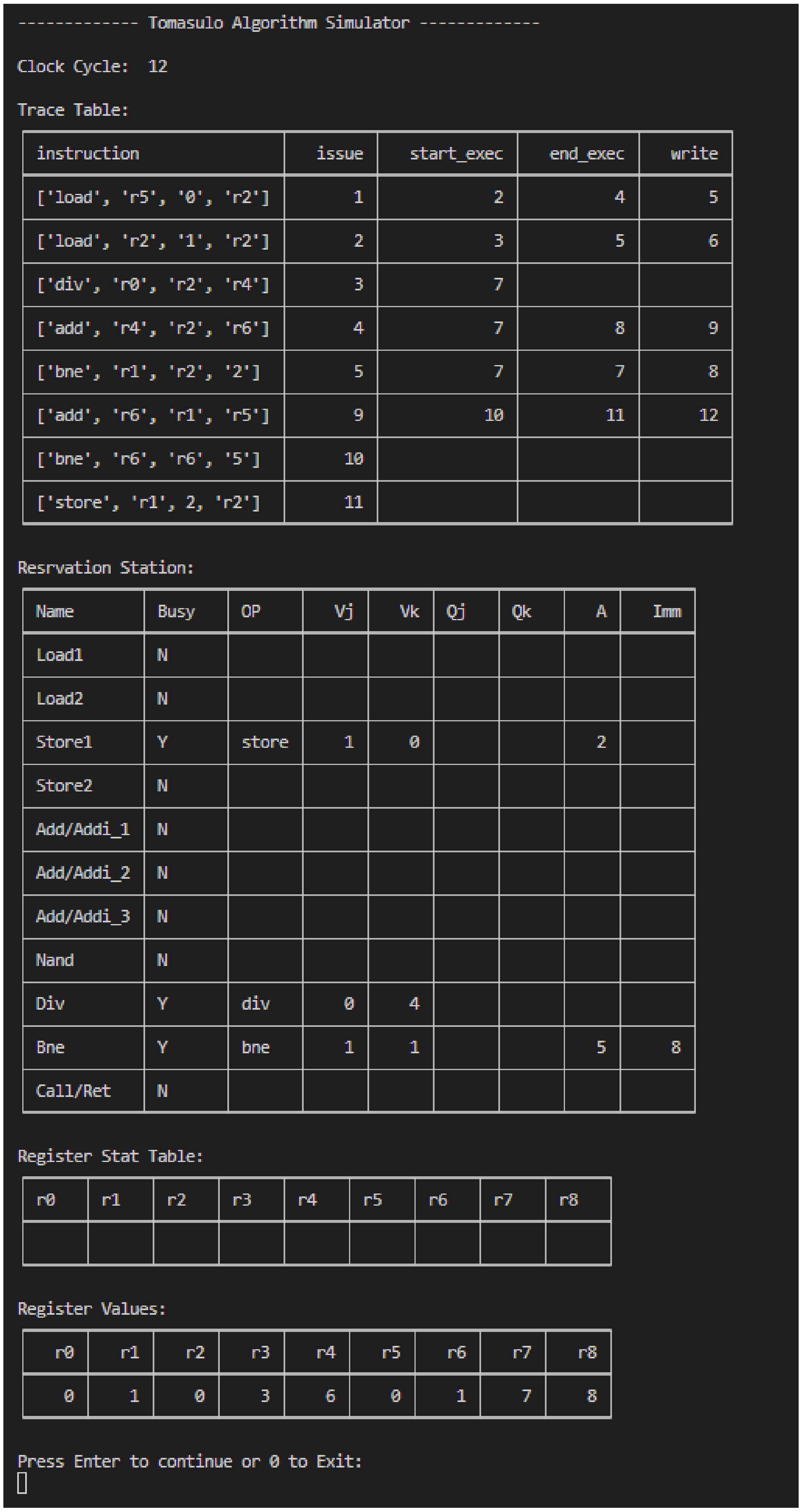
```
Press Enter to continue or 0 to Exit:
```



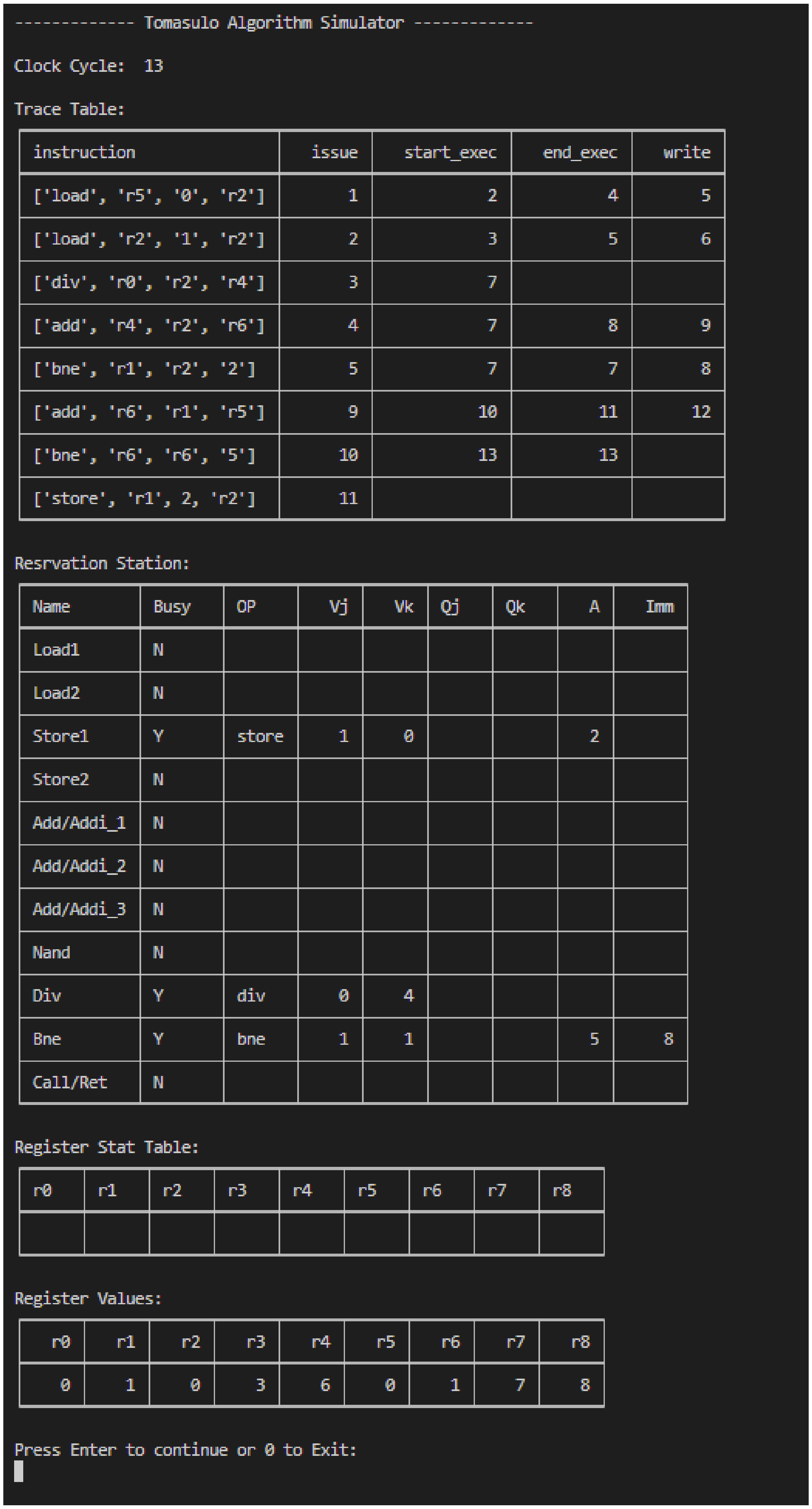
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  10

Trace Table:
| instruction                  | issue | start_exec | end_exec | write |
|------------------------------|-------|------------|----------|-------|
| ['load', 'r5', '0', 'r2']    |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']    |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']    |     3 |          7 |          |       |
| ['add', 'r4', 'r2', 'r6']    |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']     |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']    |     9 |         10 |          |       |
| ['bne', 'r6', 'r6', '5']     |    10 |            |          |       |

Resrvation Station:
| Name        | Busy | OP   | Vj | Vk | Qj          | Qk          | A | Imm |
|-------------|------|------|----|----|-------------|-------------|---|-----|
| Load1       | N    |      |    |    |             |             |   |     |
| Load2       | N    |      |    |    |             |             |   |     |
| Store1      | N    |      |    |    |             |             |   |     |
| Store2      | N    |      |    |    |             |             |   |     |
| Add/Addi_1  | N    |      |    |    |             |             |   |     |
| Add/Addi_2  | Y    | add  |  1 |  0 |             |             |   |     |
| Add/Addi_3  | N    |      |    |    |             |             |   |     |
| Nand        | N    |      |    |    |             |             |   |     |
| Div         | Y    | div  |  0 |  4 |             |             |   |     |
| Bne         | Y    | bne  |    |    | Add/Addi_2  | Add/Addi_2  | 5 |   8 |
| Call/Ret    | N    |      |    |    |             |             |   |     |

Register Stat Table:
| r0 | r1 | r2 | r3 | r4 | r5 | r6          | r7 | r8 |
|----|----|----|----|----|----|-------------|----|----|
|    |    |    |    |    |    | Add/Addi_2  |    |    |

Register Values:
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
|----|----|----|----|----|----|----|----|----|
|  0 |  1 |  0 |  3 |  6 |  0 |  6 |  7 |  8 |

Press Enter to continue or 0 to Exit:
```



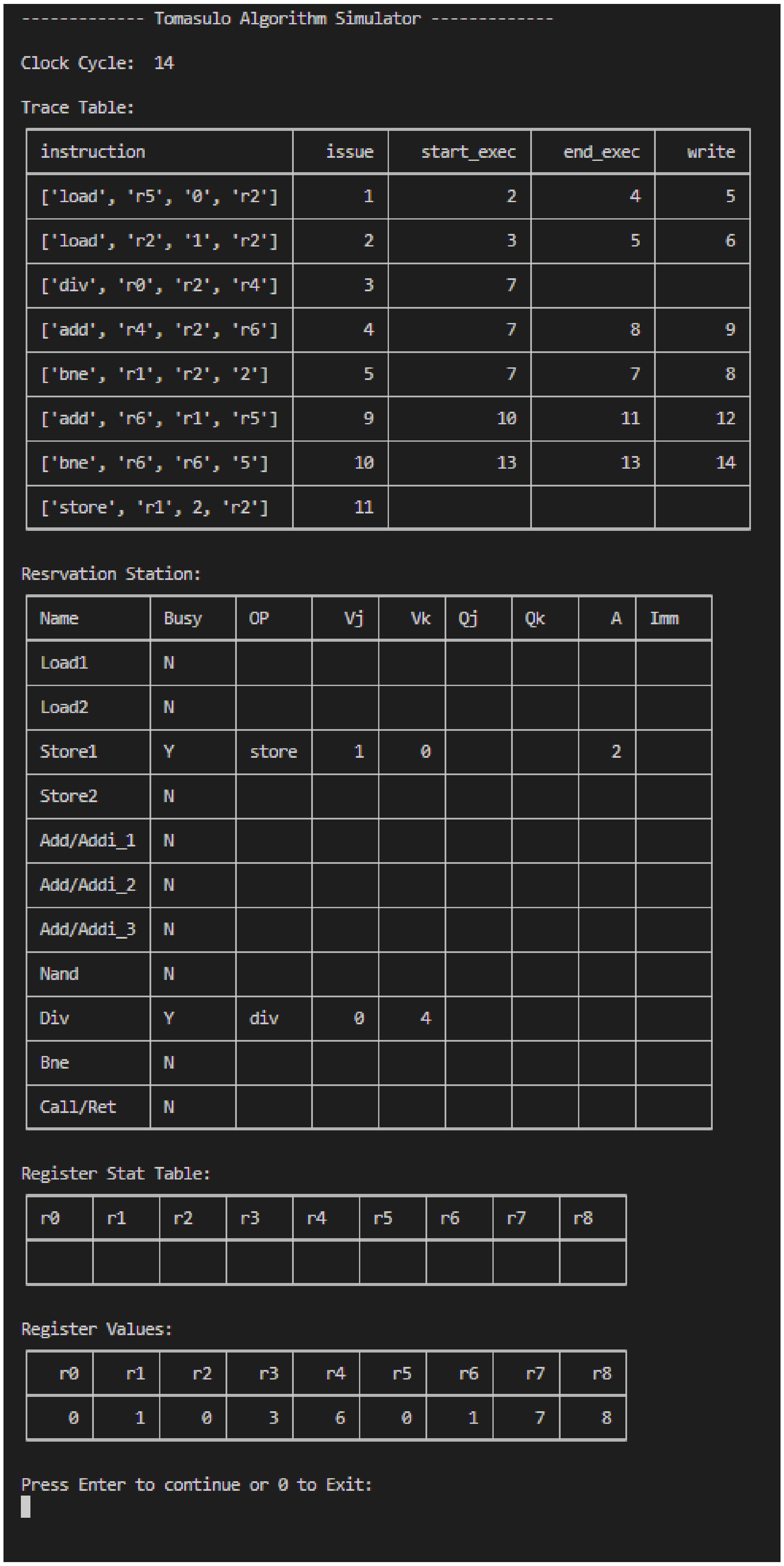
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  11

Trace Table:
+----------------------------+---------+--------------+------------+---------+
| instruction                |   issue |   start_exec |   end_exec |   write |
+----------------------------+---------+--------------+------------+---------+
| ['load', 'r5', '0', 'r2']  |       1 |            2 |          4 |       5 |
+----------------------------+---------+--------------+------------+---------+
| ['load', 'r2', '1', 'r2']  |       2 |            3 |          5 |       6 |
+----------------------------+---------+--------------+------------+---------+
| ['div', 'r0', 'r2', 'r4']  |       3 |            7 |            |         |
+----------------------------+---------+--------------+------------+---------+
| ['add', 'r4', 'r2', 'r6']  |       4 |            7 |          8 |       9 |
+----------------------------+---------+--------------+------------+---------+
| ['bne', 'r1', 'r2', '2']   |       5 |            7 |          7 |       8 |
+----------------------------+---------+--------------+------------+---------+
| ['add', 'r6', 'r1', 'r5']  |       9 |           10 |         11 |         |
+----------------------------+---------+--------------+------------+---------+
| ['bne', 'r6', 'r6', '5']   |      10 |              |            |         |
+----------------------------+---------+--------------+------------+---------+
| ['store', 'r1', 2, 'r2']   |      11 |              |            |         |
+----------------------------+---------+--------------+------------+---------+

Resrvation Station:
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Name        | Busy   | OP    |   Vj |   Vk | Qj          | Qk          |   A |   Imm |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Load1       | N      |       |      |      |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Load2       | N      |       |      |      |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Store1      | Y      | store |    1 |    0 |             |             |   2 |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Store2      | N      |       |      |      |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Add/Addi_1  | N      |       |      |      |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Add/Addi_2  | Y      | add   |    1 |    0 |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Add/Addi_3  | N      |       |      |      |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Nand        | N      |       |      |      |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Div         | Y      | div   |    0 |    4 |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Bne         | Y      | bne   |      |      | Add/Addi_2  | Add/Addi_2  |   5 |     8 |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+
| Call/Ret    | N      |       |      |      |             |             |     |       |
+-------------+--------+-------+------+------+-------------+-------------+-----+-------+

Register Stat Table:
+------+------+------+------+------+------+-------------+------+------+
| r0   | r1   | r2   | r3   | r4   | r5   | r6          | r7   | r8   |
+------+------+------+------+------+------+-------------+------+------+
|      |      |      |      |      |      | Add/Addi_2  |      |      |
+------+------+------+------+------+------+-------------+------+------+

Register Values:
+------+------+------+------+------+------+------+------+------+
|   r0 |   r1 |   r2 |   r3 |   r4 |   r5 |   r6 |   r7 |   r8 |
+------+------+------+------+------+------+------+------+------+
|    0 |    1 |    0 |    3 |    6 |    0 |    6 |    7 |    8 |
+------+------+------+------+------+------+------+------+------+

Press Enter to continue or 0 to Exit:
```



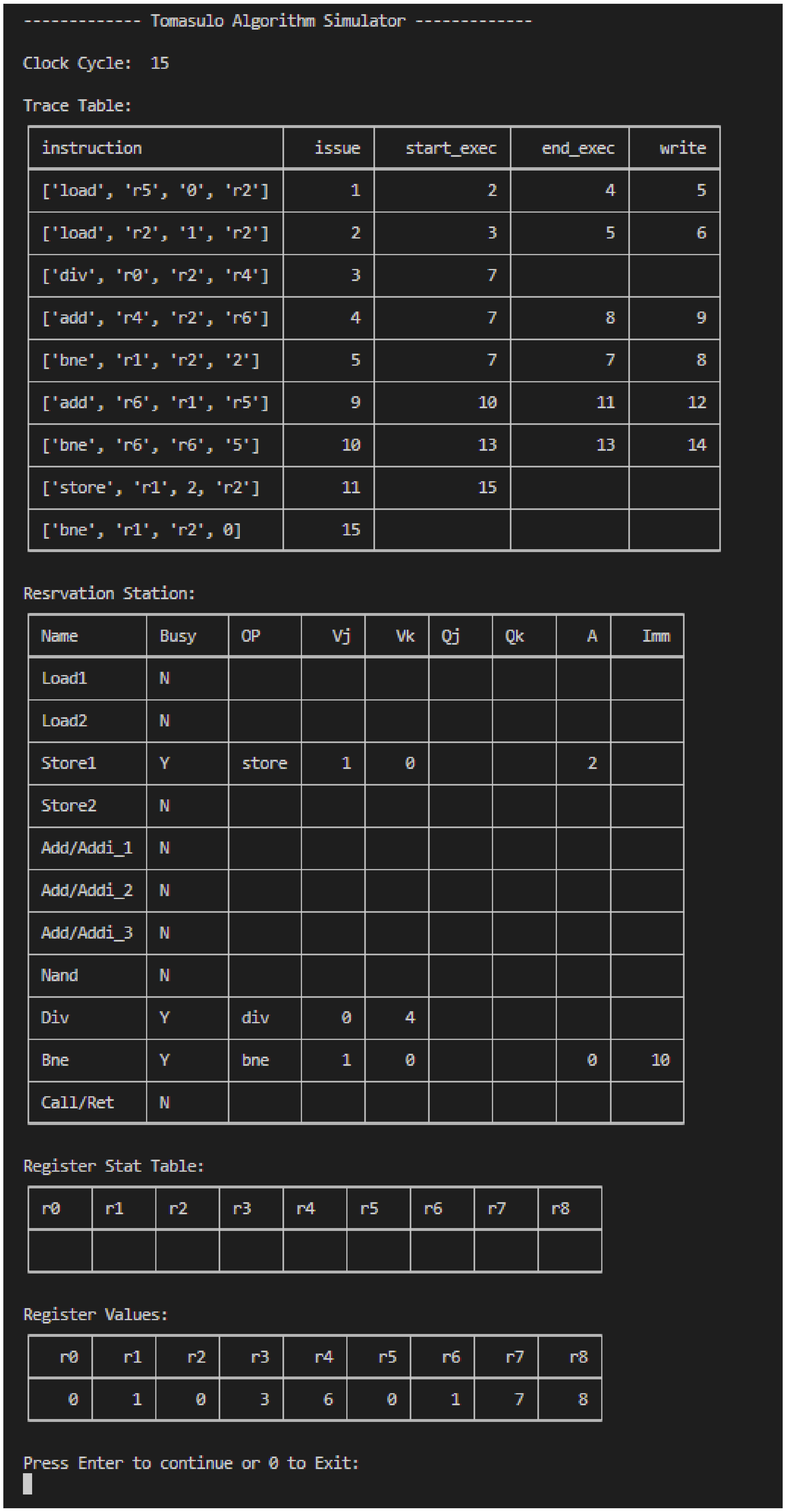
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  12

Trace Table:
+---------------------------------+---------+--------------+------------+---------+
| instruction                     |   issue |   start_exec |   end_exec |   write |
+---------------------------------+---------+--------------+------------+---------+
| ['load', 'r5', '0', 'r2']       |       1 |            2 |          4 |       5 |
| ['load', 'r2', '1', 'r2']       |       2 |            3 |          5 |       6 |
| ['div', 'r0', 'r2', 'r4']       |       3 |            7 |            |         |
| ['add', 'r4', 'r2', 'r6']       |       4 |            7 |          8 |       9 |
| ['bne', 'r1', 'r2', '2']        |       5 |            7 |          7 |       8 |
| ['add', 'r6', 'r1', 'r5']       |       9 |           10 |         11 |      12 |
| ['bne', 'r6', 'r6', '5']        |      10 |              |            |         |
| ['store', 'r1', 2, 'r2']        |      11 |              |            |         |
+---------------------------------+---------+--------------+------------+---------+

Resrvation Station:
+-------------+--------+--------+------+------+------+------+-----+-------+
| Name        | Busy   | OP     |   Vj |   Vk | Qj   | Qk   |   A |   Imm |
+-------------+--------+--------+------+------+------+------+-----+-------+
| Load1       | N      |        |      |      |      |      |     |       |
| Load2       | N      |        |      |      |      |      |     |       |
| Store1      | Y      | store  |    1 |    0 |      |      |   2 |       |
| Store2      | N      |        |      |      |      |      |     |       |
| Add/Addi_1  | N      |        |      |      |      |      |     |       |
| Add/Addi_2  | N      |        |      |      |      |      |     |       |
| Add/Addi_3  | N      |        |      |      |      |      |     |       |
| Nand        | N      |        |      |      |      |      |     |       |
| Div         | Y      | div    |    0 |    4 |      |      |     |       |
| Bne         | Y      | bne    |    1 |    1 |      |      |   5 |     8 |
| Call/Ret    | N      |        |      |      |      |      |     |       |
+-------------+--------+--------+------+------+------+------+-----+-------+

Register Stat Table:
+------+------+------+------+------+------+------+------+------+
| r0   | r1   | r2   | r3   | r4   | r5   | r6   | r7   | r8   |
+------+------+------+------+------+------+------+------+------+
|      |      |      |      |      |      |      |      |      |
+------+------+------+------+------+------+------+------+------+

Register Values:
+------+------+------+------+------+------+------+------+------+
|   r0 |   r1 |   r2 |   r3 |   r4 |   r5 |   r6 |   r7 |   r8 |
+------+------+------+------+------+------+------+------+------+
|    0 |    1 |    0 |    3 |    6 |    0 |    1 |    7 |    8 |
+------+------+------+------+------+------+------+------+------+

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  13

Trace Table:
+-----------------------------+-------+------------+----------+-------+
| instruction                 | issue | start_exec | end_exec | write |
+-----------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']   |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']   |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']   |     3 |          7 |          |       |
| ['add', 'r4', 'r2', 'r6']   |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']    |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']   |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']    |    10 |         13 |       13 |       |
| ['store', 'r1', 2, 'r2']    |    11 |            |          |       |
+-----------------------------+-------+------------+----------+-------+

Resrvation Station:
+------------+------+-------+----+----+----+----+---+-----+
| Name       | Busy | OP    | Vj | Vk | Qj | Qk | A | Imm |
+------------+------+-------+----+----+----+----+---+-----+
| Load1      | N    |       |    |    |    |    |   |     |
| Load2      | N    |       |    |    |    |    |   |     |
| Store1     | Y    | store |  1 |  0 |    |    | 2 |     |
| Store2     | N    |       |    |    |    |    |   |     |
| Add/Addi_1 | N    |       |    |    |    |    |   |     |
| Add/Addi_2 | N    |       |    |    |    |    |   |     |
| Add/Addi_3 | N    |       |    |    |    |    |   |     |
| Nand       | N    |       |    |    |    |    |   |     |
| Div        | Y    | div   |  0 |  4 |    |    |   |     |
| Bne        | Y    | bne   |  1 |  1 |    |    | 5 |   8 |
| Call/Ret   | N    |       |    |    |    |    |   |     |
+------------+------+-------+----+----+----+----+---+-----+

Register Stat Table:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 |  1 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  14

Trace Table:
| instruction                  | issue | start_exec | end_exec | write |
|------------------------------|-------|------------|----------|-------|
| ['load', 'r5', '0', 'r2']    |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']    |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']    |     3 |          7 |          |       |
| ['add', 'r4', 'r2', 'r6']    |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']     |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']    |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']     |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']     |    11 |            |          |       |

Resrvation Station:
| Name        | Busy | OP    | Vj | Vk | Qj | Qk | A | Imm |
|-------------|------|-------|----|----|----|----|---|-----|
| Load1       | N    |       |    |    |    |    |   |     |
| Load2       | N    |       |    |    |    |    |   |     |
| Store1      | Y    | store |  1 |  0 |    |    | 2 |     |
| Store2      | N    |       |    |    |    |    |   |     |
| Add/Addi_1  | N    |       |    |    |    |    |   |     |
| Add/Addi_2  | N    |       |    |    |    |    |   |     |
| Add/Addi_3  | N    |       |    |    |    |    |   |     |
| Nand        | N    |       |    |    |    |    |   |     |
| Div         | Y    | div   |  0 |  4 |    |    |   |     |
| Bne         | N    |       |    |    |    |    |   |     |
| Call/Ret    | N    |       |    |    |    |    |   |     |

Register Stat Table:
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
|----|----|----|----|----|----|----|----|----|
|    |    |    |    |    |    |    |    |    |

Register Values:
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
|----|----|----|----|----|----|----|----|----|
|  0 |  1 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  15

Trace Table:
```

instruction	issue	start_exec	end_exec	write
['load', 'r5', '0', 'r2']	1	2	4	5
['load', 'r2', '1', 'r2']	2	3	5	6
['div', 'r0', 'r2', 'r4']	3	7		
['add', 'r4', 'r2', 'r6']	4	7	8	9
['bne', 'r1', 'r2', '2']	5	7	7	8
['add', 'r6', 'r1', 'r5']	9	10	11	12
['bne', 'r6', 'r6', '5']	10	13	13	14
['store', 'r1', 2, 'r2']	11	15		
['bne', 'r1', 'r2', 0]	15			

```
Resrvation Station:
```

Name	Busy	OP	Vj	Vk	Qj	Qk	A	Imm
Load1	N							
Load2	N							
Store1	Y	store	1	0			2	
Store2	N							
Add/Addi_1	N							
Add/Addi_2	N							
Add/Addi_3	N							
Nand	N							
Div	Y	div	0	4				
Bne	Y	bne	1	0			0	10
Call/Ret	N							

```
Register Stat Table:
```

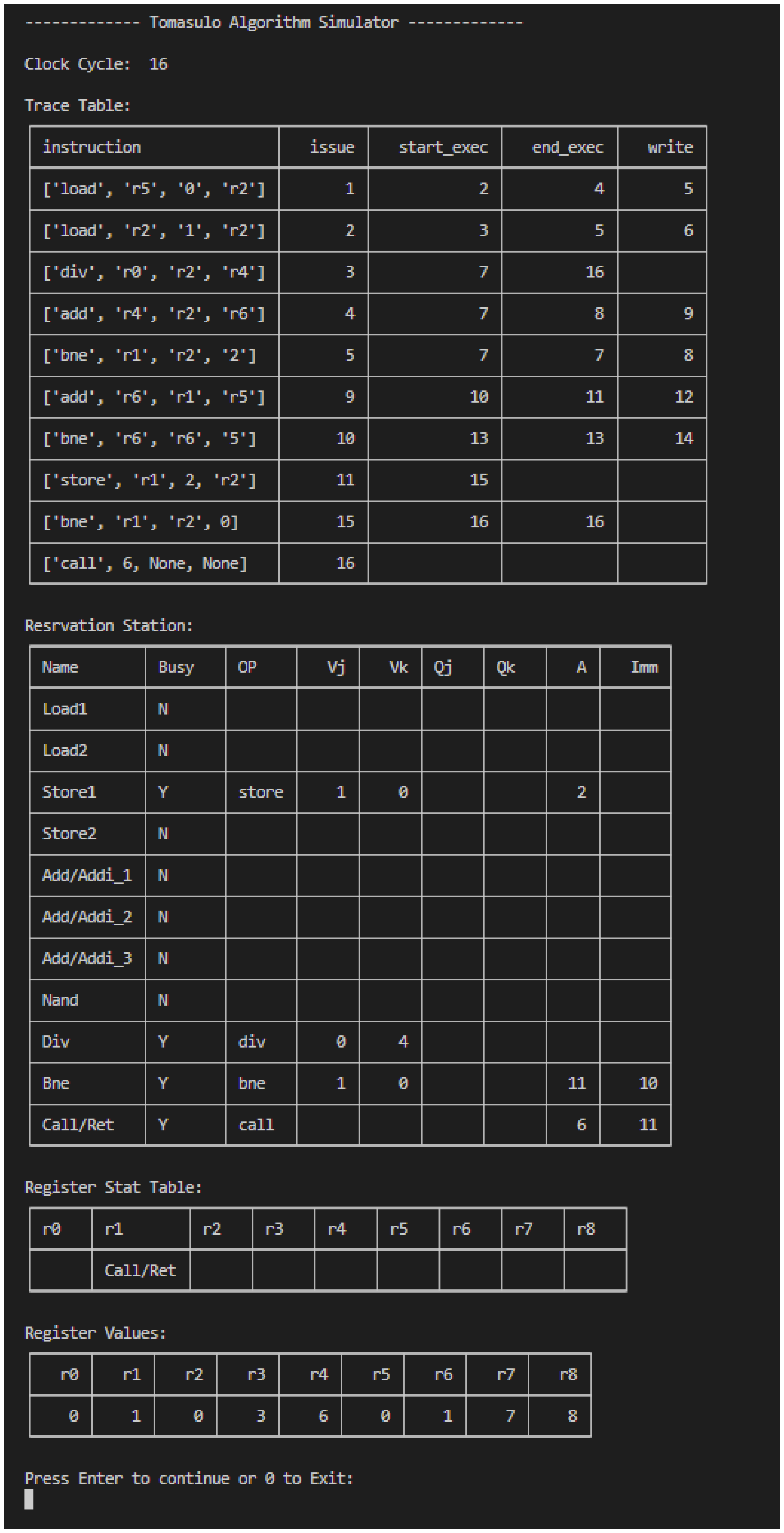
r0	r1	r2	r3	r4	r5	r6	r7	r8

```
Register Values:
```

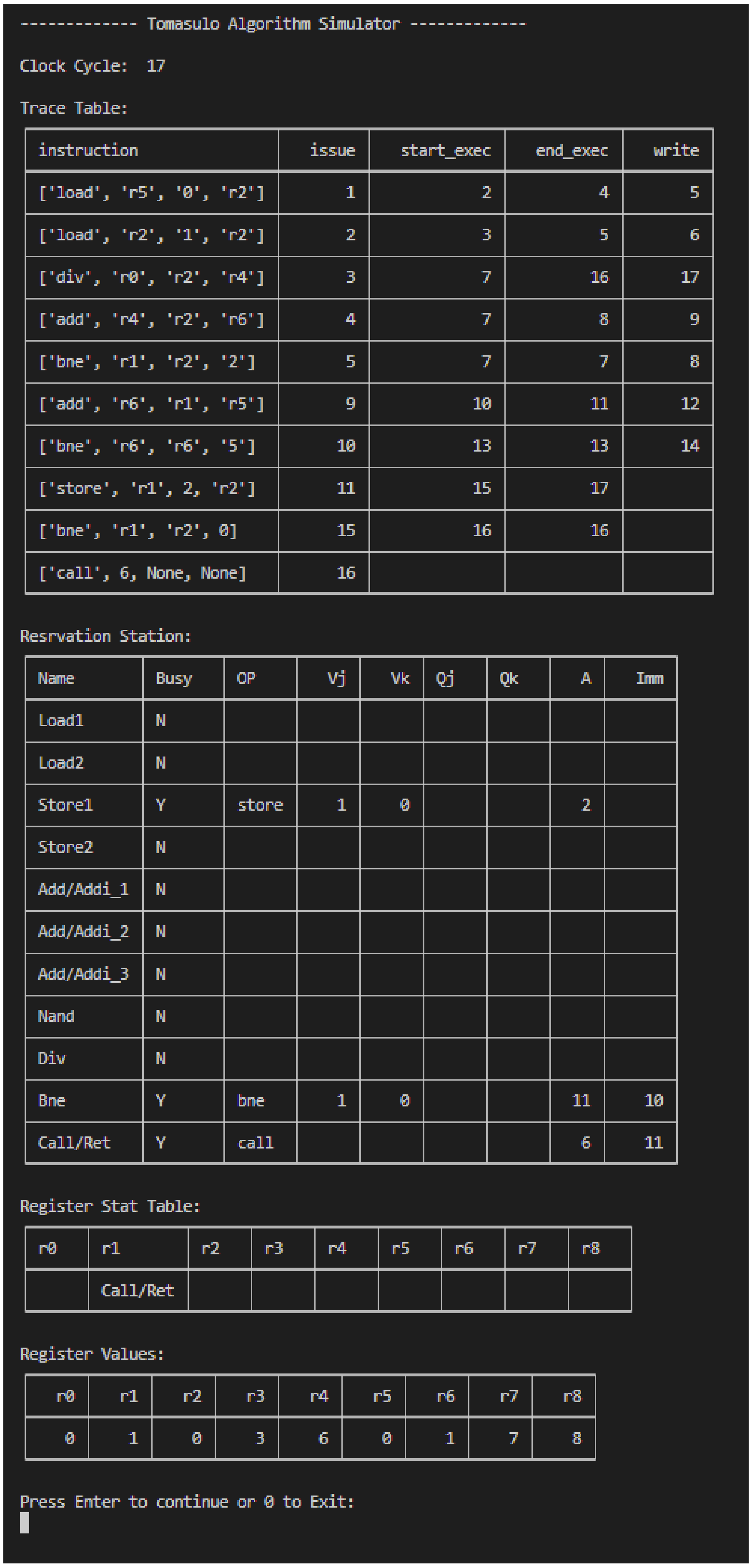
r0	r1	r2	r3	r4	r5	r6	r7	r8
0	1	0	3	6	0	1	7	8

```
Press Enter to continue or 0 to Exit:
```

Now that a call has been issued you will see that no instruction will be issued until the branch is taken

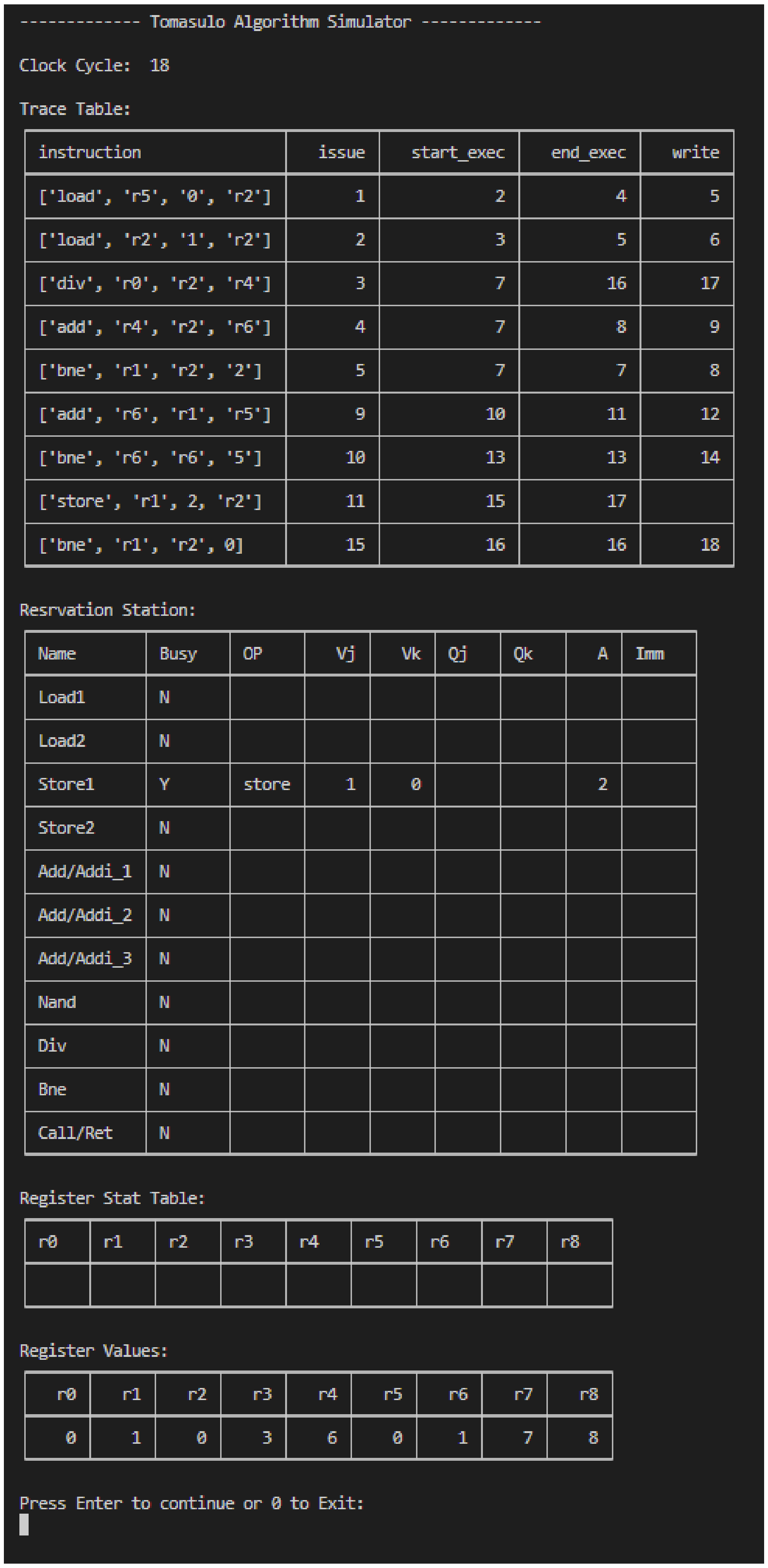


```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  16

Trace Table:
+--------------------------+-------+------------+----------+-------+
| instruction              | issue | start_exec | end_exec | write |
+--------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']|     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']|     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']|     3 |          7 |       16 |       |
| ['add', 'r4', 'r2', 'r6']|     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2'] |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']|     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5'] |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2'] |    11 |         15 |          |       |
| ['bne', 'r1', 'r2', 0]   |    15 |         16 |       16 |       |
| ['call', 6, None, None]  |    16 |            |          |       |
+--------------------------+-------+------------+----------+-------+

Resrvation Station:
+------------+------+-------+----+----+----+----+----+-----+
| Name       | Busy | OP    | Vj | Vk | Qj | Qk |  A | Imm |
+------------+------+-------+----+----+----+----+----+-----+
| Load1      | N    |       |    |    |    |    |    |     |
| Load2      | N    |       |    |    |    |    |    |     |
| Store1     | Y    | store |  1 |  0 |    |    |  2 |     |
| Store2     | N    |       |    |    |    |    |    |     |
| Add/Addi_1 | N    |       |    |    |    |    |    |     |
| Add/Addi_2 | N    |       |    |    |    |    |    |     |
| Add/Addi_3 | N    |       |    |    |    |    |    |     |
| Nand       | N    |       |    |    |    |    |    |     |
| Div        | Y    | div   |  0 |  4 |    |    |    |     |
| Bne        | Y    | bne   |  1 |  0 |    |    | 11 |  10 |
| Call/Ret   | Y    | call  |    |    |    |    |  6 |  11 |
+------------+------+-------+----+----+----+----+----+-----+

Register Stat Table:
+----+----------+----+----+----+----+----+----+----+
| r0 | r1       | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----------+----+----+----+----+----+----+----+
|    | Call/Ret |    |    |    |    |    |    |    |
+----+----------+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 |  1 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```

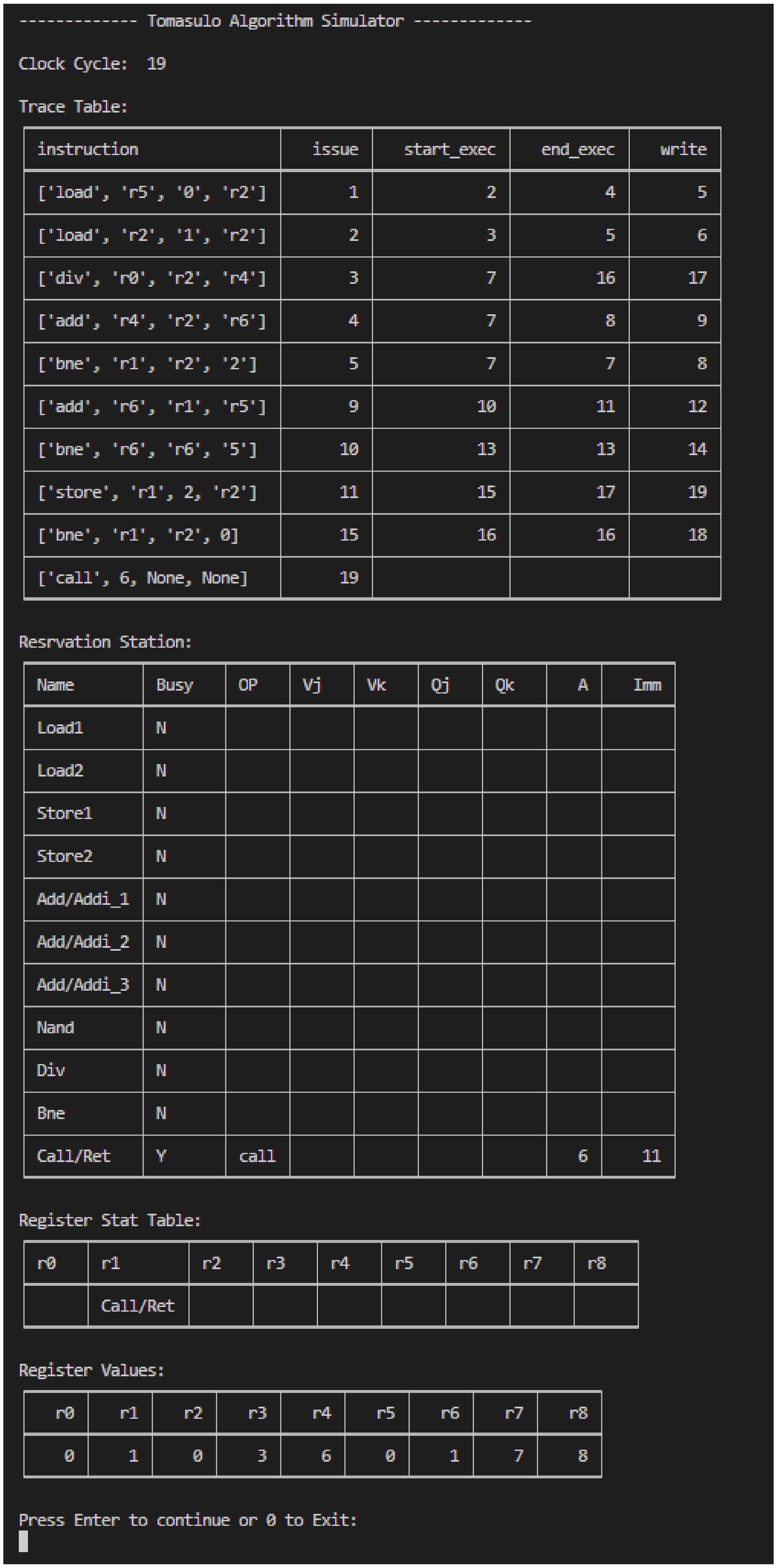


```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  17

Trace Table:
+--------------------------------+-------+------------+----------+-------+
| instruction                    | issue | start_exec | end_exec | write |
+--------------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']      |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']      |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']      |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']      |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']       |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']      |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']       |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']       |    11 |         15 |       17 |       |
| ['bne', 'r1', 'r2', 0]         |    15 |         16 |       16 |       |
| ['call', 6, None, None]        |    16 |            |          |       |
+--------------------------------+-------+------------+----------+-------+

Resrvation Station:
+------------+------+-------+----+----+----+----+----+-----+
| Name       | Busy | OP    | Vj | Vk | Qj | Qk |  A | Imm |
+------------+------+-------+----+----+----+----+----+-----+
| Load1      | N    |       |    |    |    |    |    |     |
| Load2      | N    |       |    |    |    |    |    |     |
| Store1     | Y    | store |  1 |  0 |    |    |  2 |     |
| Store2     | N    |       |    |    |    |    |    |     |
| Add/Addi_1 | N    |       |    |    |    |    |    |     |
| Add/Addi_2 | N    |       |    |    |    |    |    |     |
| Add/Addi_3 | N    |       |    |    |    |    |    |     |
| Nand       | N    |       |    |    |    |    |    |     |
| Div        | N    |       |    |    |    |    |    |     |
| Bne        | Y    | bne   |  1 |  0 |    |    | 11 |  10 |
| Call/Ret   | Y    | call  |    |    |    |    |  6 |  11 |
+------------+------+-------+----+----+----+----+----+-----+

Register Stat Table:
+----+----------+----+----+----+----+----+----+----+
| r0 | r1       | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----------+----+----+----+----+----+----+----+
|    | Call/Ret |    |    |    |    |    |    |    |
+----+----------+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 |  1 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```

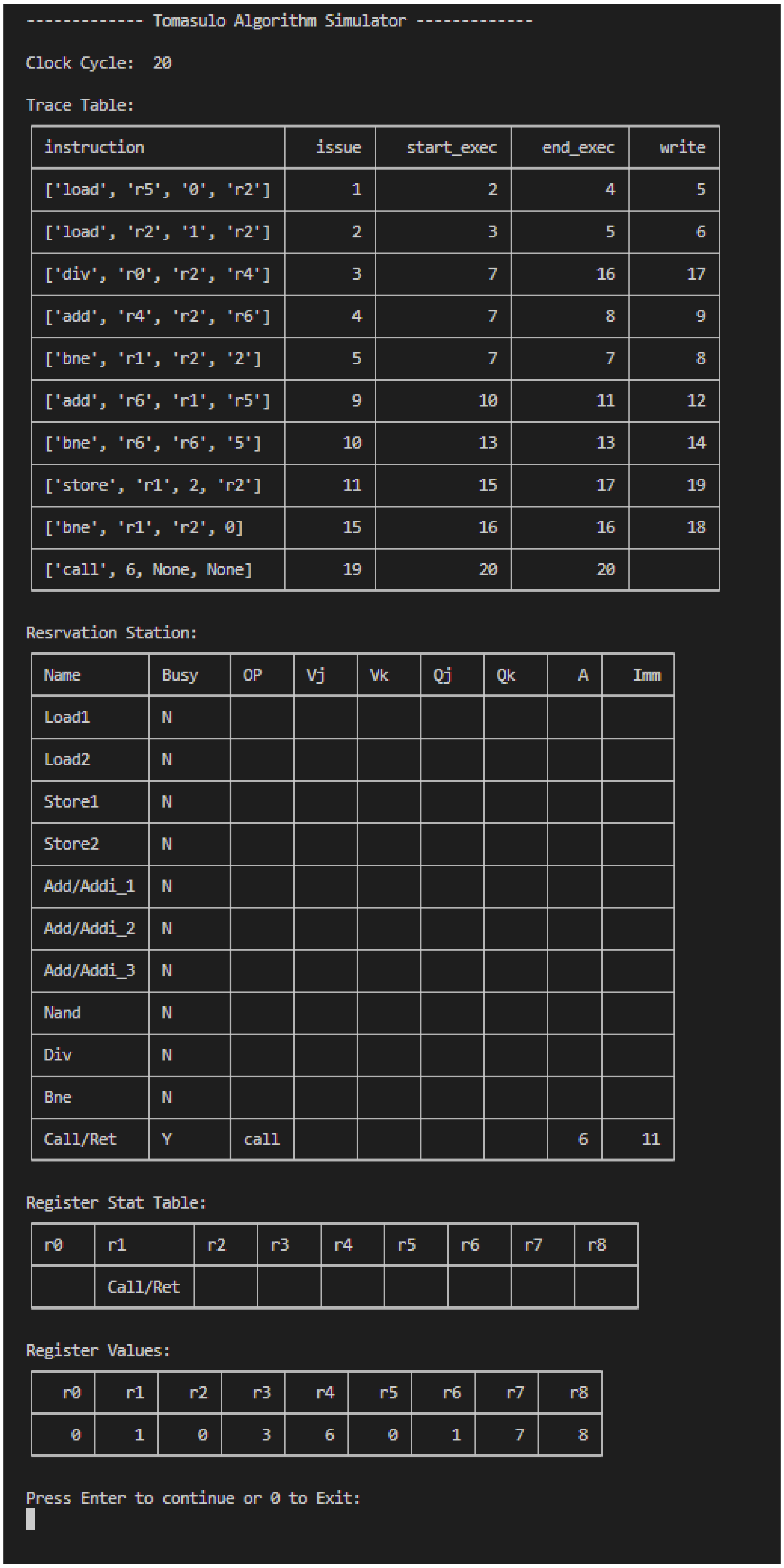
In this Cycle we see that the branch will be taken and so we flush the instructions from the trace table



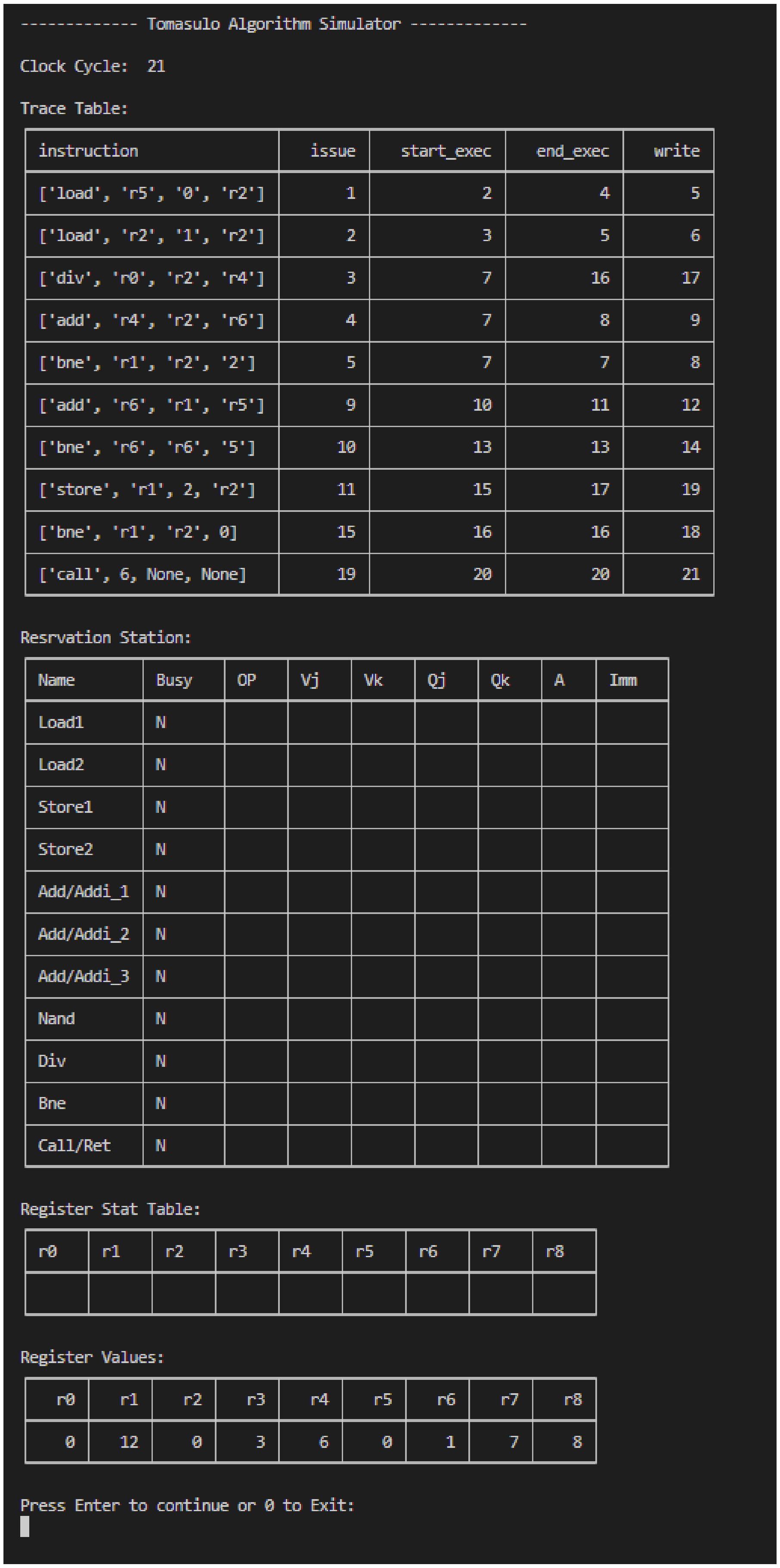
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  18

Trace Table:
+----------------------------+---------+-------------+-----------+---------+
| instruction                |   issue |   start_exec |  end_exec |   write |
+----------------------------+---------+-------------+-----------+---------+
| ['load', 'r5', '0', 'r2']  |       1 |           2 |         4 |       5 |
| ['load', 'r2', '1', 'r2']  |       2 |           3 |         5 |       6 |
| ['div', 'r0', 'r2', 'r4']  |       3 |           7 |        16 |      17 |
| ['add', 'r4', 'r2', 'r6']  |       4 |           7 |         8 |       9 |
| ['bne', 'r1', 'r2', '2']   |       5 |           7 |         7 |       8 |
| ['add', 'r6', 'r1', 'r5']  |       9 |          10 |        11 |      12 |
| ['bne', 'r6', 'r6', '5']   |      10 |          13 |        13 |      14 |
| ['store', 'r1', 2, 'r2']   |      11 |          15 |        17 |         |
| ['bne', 'r1', 'r2', 0]     |      15 |          16 |        16 |      18 |
+----------------------------+---------+-------------+-----------+---------+

Resrvation Station:
+------------+--------+-------+------+------+------+------+-----+-------+
| Name       | Busy   | OP    |   Vj |   Vk | Qj   | Qk   |   A | Imm   |
+------------+--------+-------+------+------+------+------+-----+-------+
| Load1      | N      |       |      |      |      |      |     |       |
| Load2      | N      |       |      |      |      |      |     |       |
| Store1     | Y      | store |    1 |    0 |      |      |   2 |       |
| Store2     | N      |       |      |      |      |      |     |       |
| Add/Addi_1 | N      |       |      |      |      |      |     |       |
| Add/Addi_2 | N      |       |      |      |      |      |     |       |
| Add/Addi_3 | N      |       |      |      |      |      |     |       |
| Nand       | N      |       |      |      |      |      |     |       |
| Div        | N      |       |      |      |      |      |     |       |
| Bne        | N      |       |      |      |      |      |     |       |
| Call/Ret   | N      |       |      |      |      |      |     |       |
+------------+--------+-------+------+------+------+------+-----+-------+

Register Stat Table:
+------+------+------+------+------+------+------+------+------+
| r0   | r1   | r2   | r3   | r4   | r5   | r6   | r7   | r8   |
+------+------+------+------+------+------+------+------+------+
|      |      |      |      |      |      |      |      |      |
+------+------+------+------+------+------+------+------+------+

Register Values:
+------+------+------+------+------+------+------+------+------+
|   r0 |   r1 |   r2 |   r3 |   r4 |   r5 |   r6 |   r7 |   r8 |
+------+------+------+------+------+------+------+------+------+
|    0 |    1 |    0 |    3 |    6 |    0 |    1 |    7 |    8 |
+------+------+------+------+------+------+------+------+------+

Press Enter to continue or 0 to Exit:
```



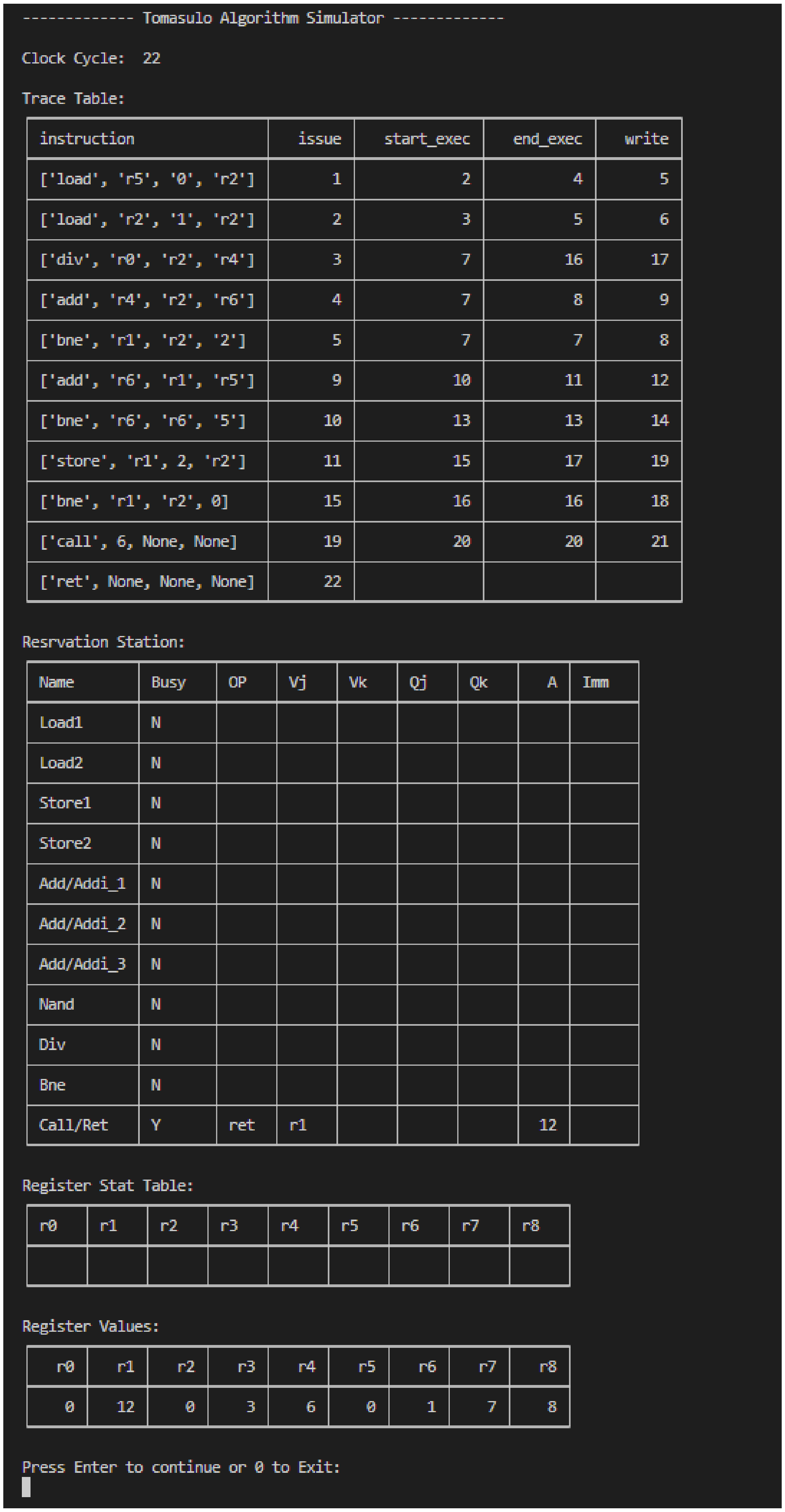
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  19

Trace Table:
+----------------------------+-------+------------+----------+-------+
| instruction                | issue | start_exec | end_exec | write |
+----------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']  |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']  |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']  |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']  |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']   |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']  |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']   |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']   |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]     |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]    |    19 |            |          |       |
+----------------------------+-------+------------+----------+-------+

Resrvation Station:
+------------+------+------+----+----+----+----+---+-----+
| Name       | Busy | OP   | Vj | Vk | Qj | Qk | A | Imm |
+------------+------+------+----+----+----+----+---+-----+
| Load1      | N    |      |    |    |    |    |   |     |
| Load2      | N    |      |    |    |    |    |   |     |
| Store1     | N    |      |    |    |    |    |   |     |
| Store2     | N    |      |    |    |    |    |   |     |
| Add/Addi_1 | N    |      |    |    |    |    |   |     |
| Add/Addi_2 | N    |      |    |    |    |    |   |     |
| Add/Addi_3 | N    |      |    |    |    |    |   |     |
| Nand       | N    |      |    |    |    |    |   |     |
| Div        | N    |      |    |    |    |    |   |     |
| Bne        | N    |      |    |    |    |    |   |     |
| Call/Ret   | Y    | call |    |    |    |    | 6 |  11 |
+------------+------+------+----+----+----+----+---+-----+

Register Stat Table:
+----+----------+----+----+----+----+----+----+----+
| r0 | r1       | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----------+----+----+----+----+----+----+----+
|    | Call/Ret |    |    |    |    |    |    |    |
+----+----------+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 |  1 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
█
```



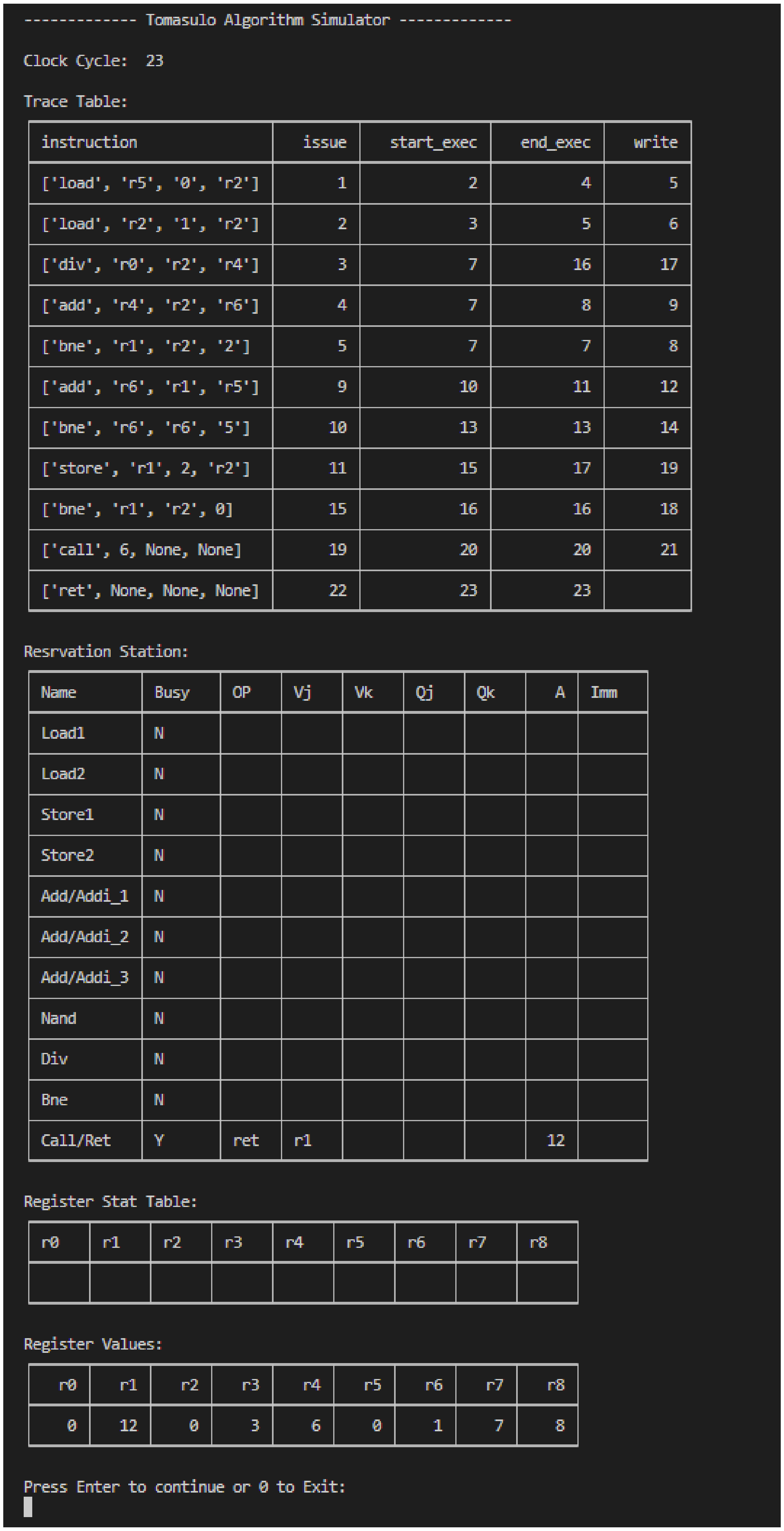
```
------------ Tomasulo Algorithm Simulator ------------

Clock Cycle:  20

Trace Table:
| instruction                   | issue | start_exec | end_exec | write |
| ['load', 'r5', '0', 'r2']     |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']     |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']     |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']     |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']      |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']     |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']      |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']      |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]        |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]       |    19 |         20 |       20 |       |

Resrvation Station:
| Name       | Busy | OP   | Vj | Vk | Qj | Qk | A | Imm |
| Load1      | N    |      |    |    |    |    |   |     |
| Load2      | N    |      |    |    |    |    |   |     |
| Store1     | N    |      |    |    |    |    |   |     |
| Store2     | N    |      |    |    |    |    |   |     |
| Add/Addi_1 | N    |      |    |    |    |    |   |     |
| Add/Addi_2 | N    |      |    |    |    |    |   |     |
| Add/Addi_3 | N    |      |    |    |    |    |   |     |
| Nand       | N    |      |    |    |    |    |   |     |
| Div        | N    |      |    |    |    |    |   |     |
| Bne        | N    |      |    |    |    |    |   |     |
| Call/Ret   | Y    | call |    |    |    |    | 6 |  11 |

Register Stat Table:
| r0 | r1       | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
|    | Call/Ret |    |    |    |    |    |    |    |

Register Values:
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
|  0 |  1 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |

Press Enter to continue or 0 to Exit:
```



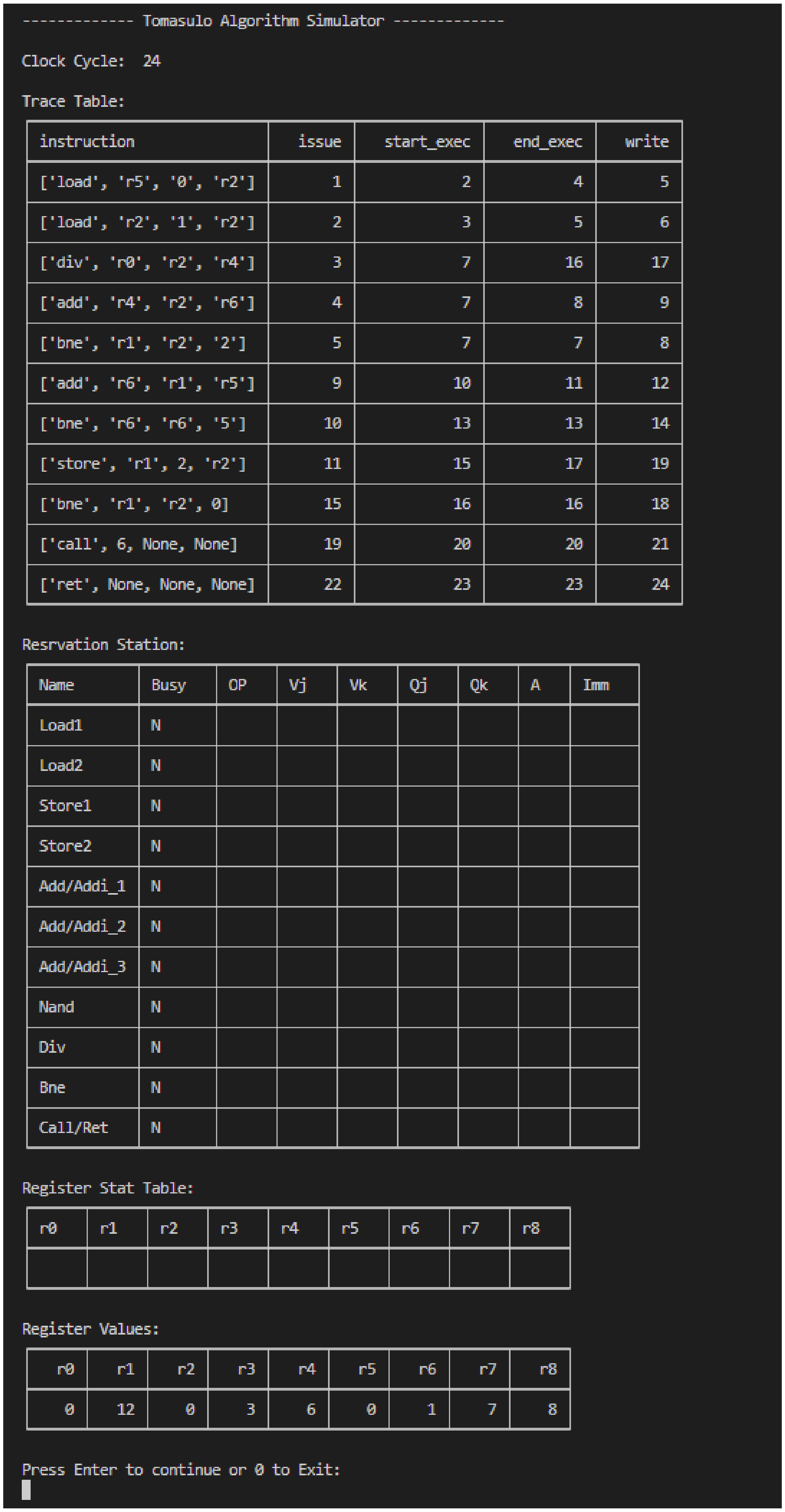
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  21

Trace Table:
| instruction                  | issue | start_exec | end_exec | write |
| ['load', 'r5', '0', 'r2']    |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']    |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']    |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']    |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']     |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']    |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']     |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']     |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]       |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]      |    19 |         20 |       20 |    21 |

Resrvation Station:
| Name        | Busy | OP | Vj | Vk | Qj | Qk | A | Imm |
| Load1       | N    |    |    |    |    |    |   |     |
| Load2       | N    |    |    |    |    |    |   |     |
| Store1      | N    |    |    |    |    |    |   |     |
| Store2      | N    |    |    |    |    |    |   |     |
| Add/Addi_1  | N    |    |    |    |    |    |   |     |
| Add/Addi_2  | N    |    |    |    |    |    |   |     |
| Add/Addi_3  | N    |    |    |    |    |    |   |     |
| Nand        | N    |    |    |    |    |    |   |     |
| Div         | N    |    |    |    |    |    |   |     |
| Bne         | N    |    |    |    |    |    |   |     |
| Call/Ret    | N    |    |    |    |    |    |   |     |

Register Stat Table:
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
|    |    |    |    |    |    |    |    |    |

Register Values:
|  r0 |  r1 |  r2 |  r3 |  r4 |  r5 |  r6 |  r7 |  r8 |
|   0 |  12 |   0 |   3 |   6 |   0 |   1 |   7 |   8 |

Press Enter to continue or 0 to Exit:
```



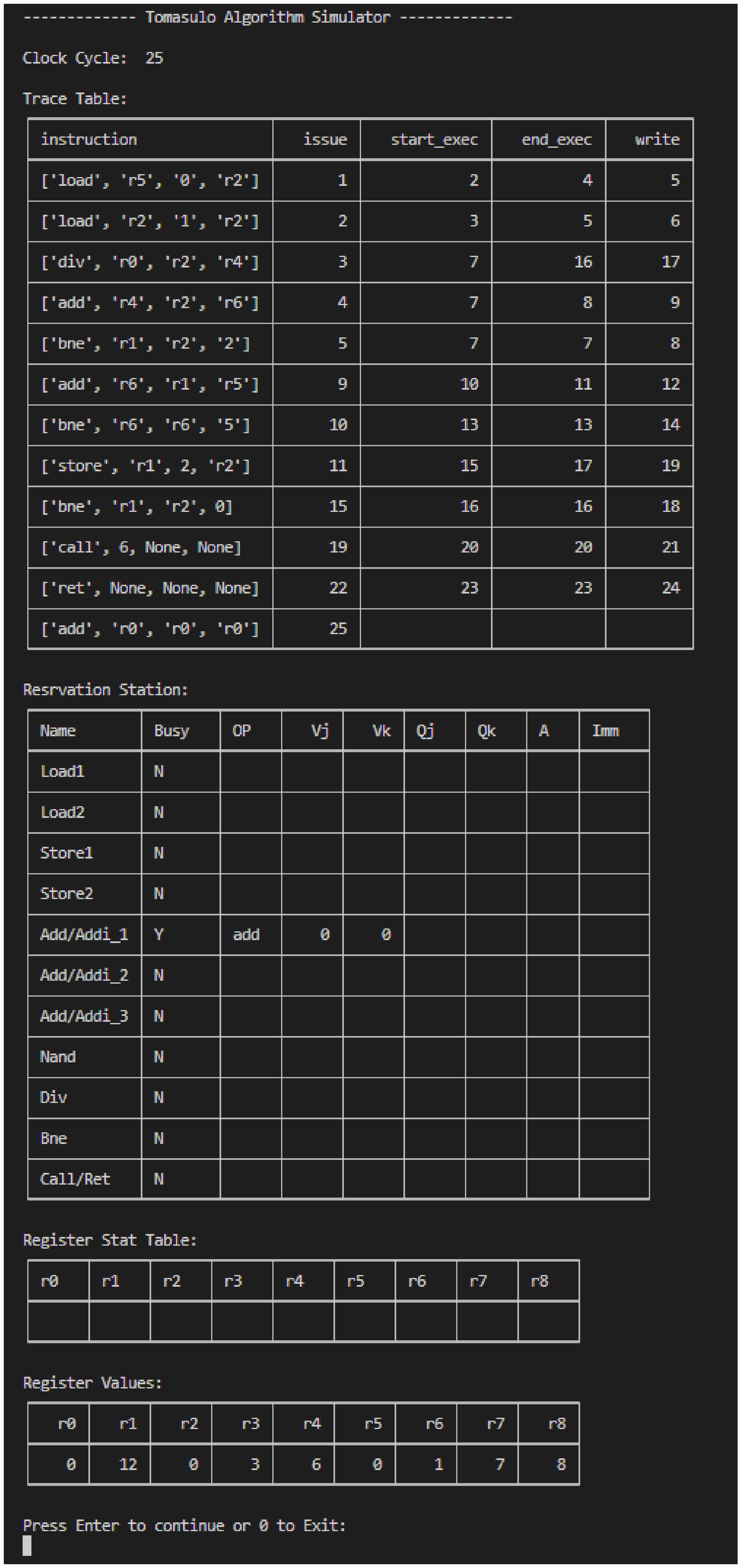
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  22

Trace Table:
+----------------------------------+-------+------------+----------+-------+
| instruction                      | issue | start_exec | end_exec | write |
+----------------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']        |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']        |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']        |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']        |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']         |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']        |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']         |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']         |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]           |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]          |    19 |         20 |       20 |    21 |
| ['ret', None, None, None]        |    22 |            |          |       |
+----------------------------------+-------+------------+----------+-------+

Resrvation Station:
+------------+------+-----+----+----+----+----+----+-----+
| Name       | Busy | OP  | Vj | Vk | Qj | Qk |  A | Imm |
+------------+------+-----+----+----+----+----+----+-----+
| Load1      | N    |     |    |    |    |    |    |     |
| Load2      | N    |     |    |    |    |    |    |     |
| Store1     | N    |     |    |    |    |    |    |     |
| Store2     | N    |     |    |    |    |    |    |     |
| Add/Addi_1 | N    |     |    |    |    |    |    |     |
| Add/Addi_2 | N    |     |    |    |    |    |    |     |
| Add/Addi_3 | N    |     |    |    |    |    |    |     |
| Nand       | N    |     |    |    |    |    |    |     |
| Div        | N    |     |    |    |    |    |    |     |
| Bne        | N    |     |    |    |    |    |    |     |
| Call/Ret   | Y    | ret | r1 |    |    |    | 12 |     |
+------------+------+-----+----+----+----+----+----+-----+

Register Stat Table:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 | 12 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```



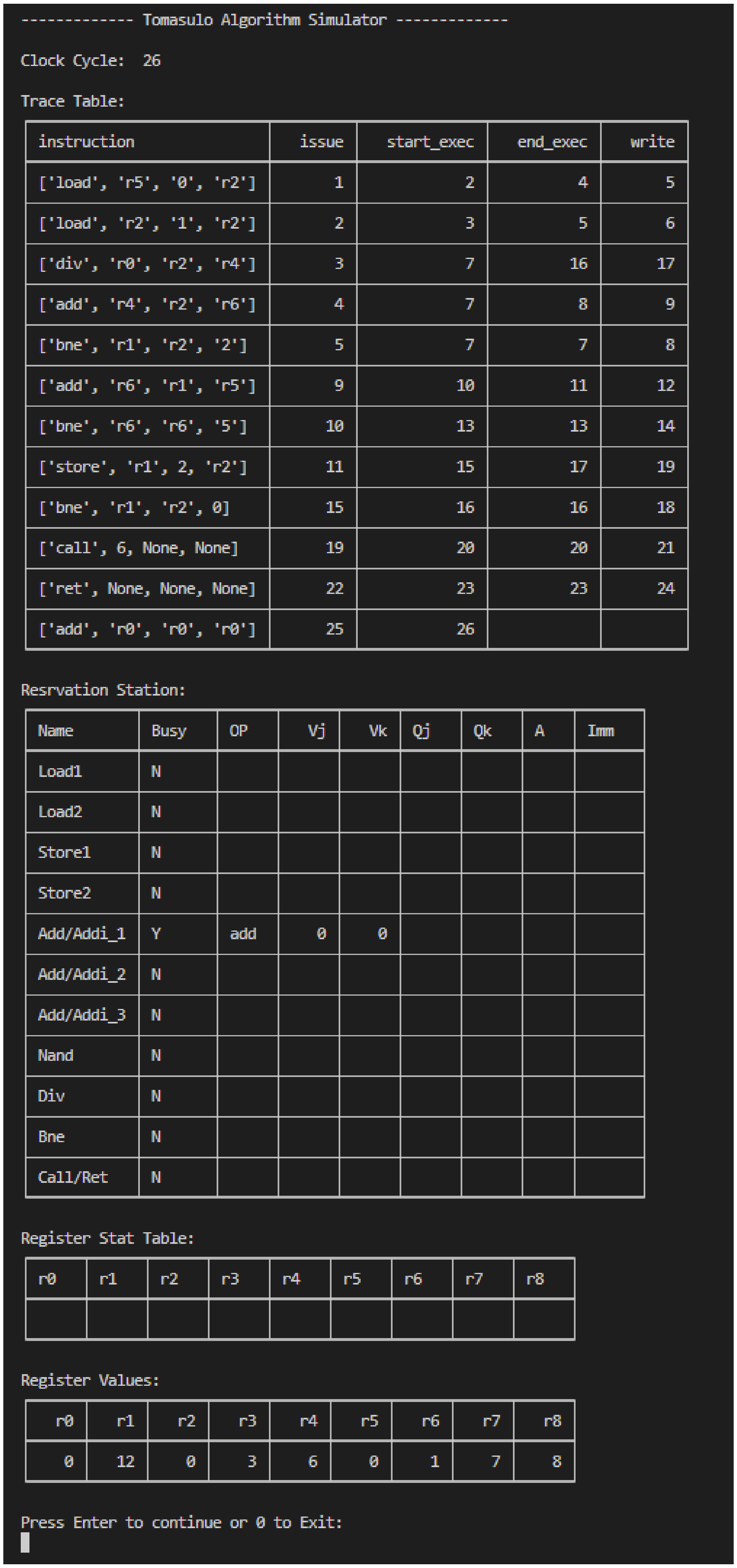
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  23

Trace Table:
+--------------------------------+-------+------------+----------+-------+
| instruction                    | issue | start_exec | end_exec | write |
+--------------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']      |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']      |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']      |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']      |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']       |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']      |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']       |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']       |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]         |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]        |    19 |         20 |       20 |    21 |
| ['ret', None, None, None]      |    22 |         23 |       23 |       |
+--------------------------------+-------+------------+----------+-------+

Resrvation Station:
+-------------+------+-----+----+----+----+----+----+-----+
| Name        | Busy | OP  | Vj | Vk | Qj | Qk |  A | Imm |
+-------------+------+-----+----+----+----+----+----+-----+
| Load1       | N    |     |    |    |    |    |    |     |
| Load2       | N    |     |    |    |    |    |    |     |
| Store1      | N    |     |    |    |    |    |    |     |
| Store2      | N    |     |    |    |    |    |    |     |
| Add/Addi_1  | N    |     |    |    |    |    |    |     |
| Add/Addi_2  | N    |     |    |    |    |    |    |     |
| Add/Addi_3  | N    |     |    |    |    |    |    |     |
| Nand        | N    |     |    |    |    |    |    |     |
| Div         | N    |     |    |    |    |    |    |     |
| Bne         | N    |     |    |    |    |    |    |     |
| Call/Ret    | Y    | ret | r1 |    |    |    | 12 |     |
+-------------+------+-----+----+----+----+----+----+-----+

Register Stat Table:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 | 12 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```



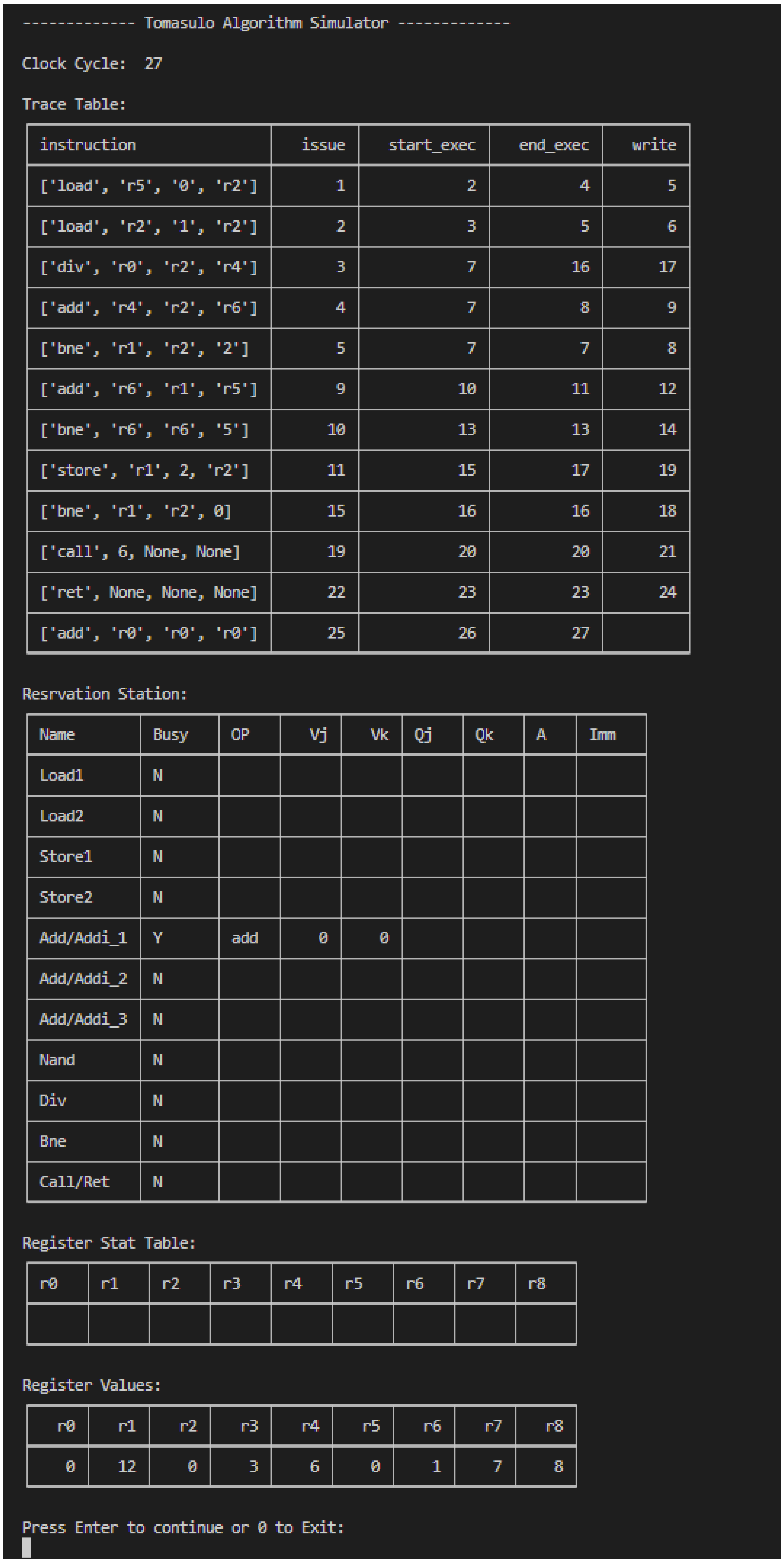
```
------------ Tomasulo Algorithm Simulator ------------

Clock Cycle:  24

Trace Table:
+----------------------------------+-------+------------+----------+-------+
| instruction                      | issue | start_exec | end_exec | write |
+----------------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']        |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']        |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']        |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']        |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']         |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']        |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']         |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']         |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]           |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]          |    19 |         20 |       20 |    21 |
| ['ret', None, None, None]        |    22 |         23 |       23 |    24 |
+----------------------------------+-------+------------+----------+-------+

Resrvation Station:
+-------------+------+----+----+----+----+----+---+-----+
| Name        | Busy | OP | Vj | Vk | Qj | Qk | A | Imm |
+-------------+------+----+----+----+----+----+---+-----+
| Load1       | N    |    |    |    |    |    |   |     |
| Load2       | N    |    |    |    |    |    |   |     |
| Store1      | N    |    |    |    |    |    |   |     |
| Store2      | N    |    |    |    |    |    |   |     |
| Add/Addi_1  | N    |    |    |    |    |    |   |     |
| Add/Addi_2  | N    |    |    |    |    |    |   |     |
| Add/Addi_3  | N    |    |    |    |    |    |   |     |
| Nand        | N    |    |    |    |    |    |   |     |
| Div         | N    |    |    |    |    |    |   |     |
| Bne         | N    |    |    |    |    |    |   |     |
| Call/Ret    | N    |    |    |    |    |    |   |     |
+-------------+------+----+----+----+----+----+---+-----+

Register Stat Table:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 | 12 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```



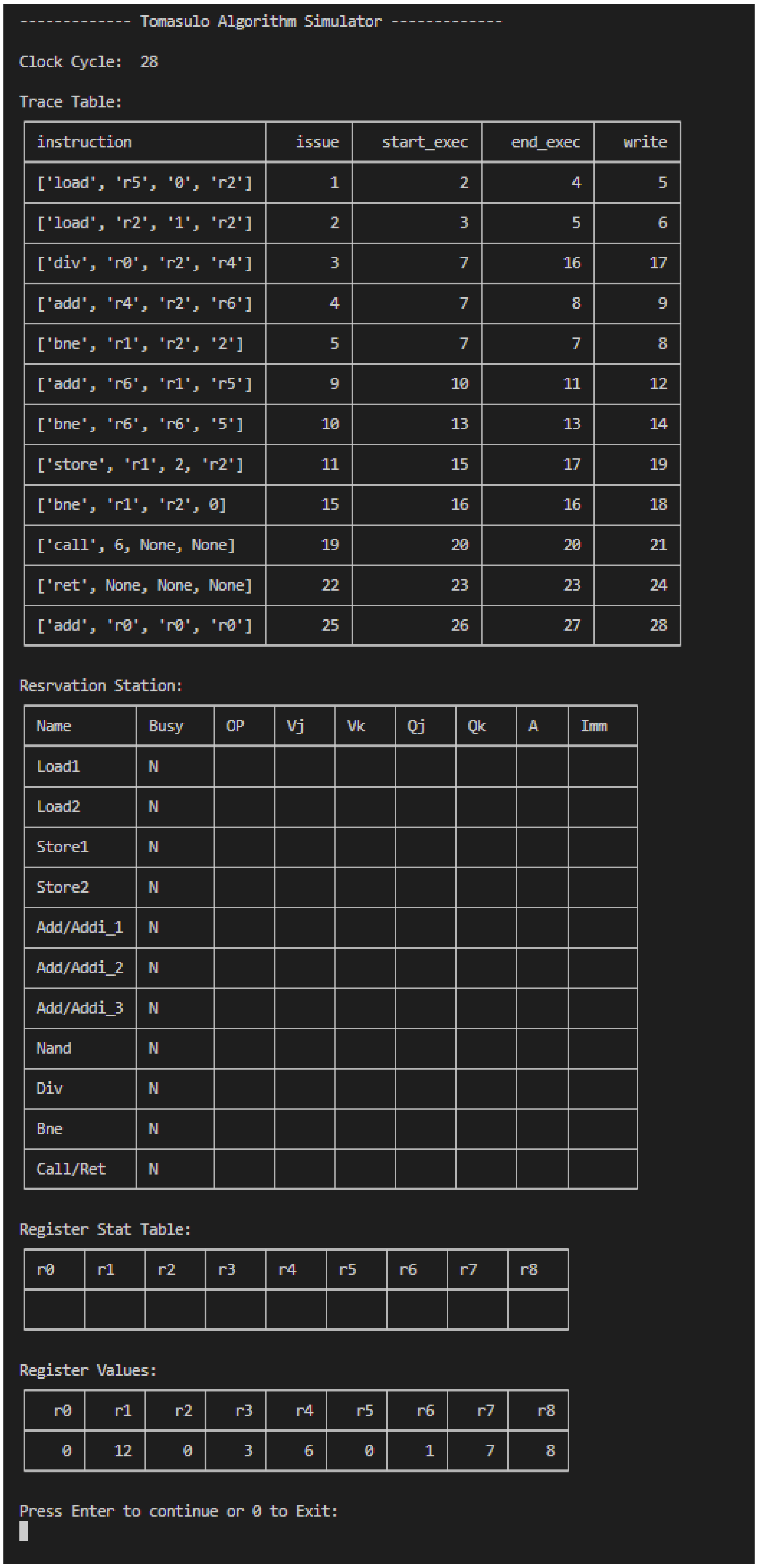
```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  25

Trace Table:
+--------------------------------+-------+------------+----------+-------+
| instruction                    | issue | start_exec | end_exec | write |
+--------------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']      |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']      |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']      |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']      |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']       |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']      |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']       |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']       |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]         |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]        |    19 |         20 |       20 |    21 |
| ['ret', None, None, None]      |    22 |         23 |       23 |    24 |
| ['add', 'r0', 'r0', 'r0']      |    25 |            |          |       |
+--------------------------------+-------+------------+----------+-------+

Resrvation Station:
+-------------+------+-----+----+----+----+----+---+-----+
| Name        | Busy | OP  | Vj | Vk | Qj | Qk | A | Imm |
+-------------+------+-----+----+----+----+----+---+-----+
| Load1       | N    |     |    |    |    |    |   |     |
| Load2       | N    |     |    |    |    |    |   |     |
| Store1      | N    |     |    |    |    |    |   |     |
| Store2      | N    |     |    |    |    |    |   |     |
| Add/Addi_1  | Y    | add |  0 |  0 |    |    |   |     |
| Add/Addi_2  | N    |     |    |    |    |    |   |     |
| Add/Addi_3  | N    |     |    |    |    |    |   |     |
| Nand        | N    |     |    |    |    |    |   |     |
| Div         | N    |     |    |    |    |    |   |     |
| Bne         | N    |     |    |    |    |    |   |     |
| Call/Ret    | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+

Register Stat Table:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 | 12 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  26

Trace Table:
+------------------------------+-------+------------+----------+-------+
| instruction                  | issue | start_exec | end_exec | write |
+------------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']    |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']    |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']    |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']    |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']     |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']    |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']     |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']     |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]       |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]      |    19 |         20 |       20 |    21 |
| ['ret', None, None, None]    |    22 |         23 |       23 |    24 |
| ['add', 'r0', 'r0', 'r0']    |    25 |         26 |          |       |
+------------------------------+-------+------------+----------+-------+

Resrvation Station:
+-------------+------+-----+----+----+----+----+---+-----+
| Name        | Busy | OP  | Vj | Vk | Qj | Qk | A | Imm |
+-------------+------+-----+----+----+----+----+---+-----+
| Load1       | N    |     |    |    |    |    |   |     |
| Load2       | N    |     |    |    |    |    |   |     |
| Store1      | N    |     |    |    |    |    |   |     |
| Store2      | N    |     |    |    |    |    |   |     |
| Add/Addi_1  | Y    | add |  0 |  0 |    |    |   |     |
| Add/Addi_2  | N    |     |    |    |    |    |   |     |
| Add/Addi_3  | N    |     |    |    |    |    |   |     |
| Nand        | N    |     |    |    |    |    |   |     |
| Div         | N    |     |    |    |    |    |   |     |
| Bne         | N    |     |    |    |    |    |   |     |
| Call/Ret    | N    |     |    |    |    |    |   |     |
+-------------+------+-----+----+----+----+----+---+-----+

Register Stat Table:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 | 12 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```

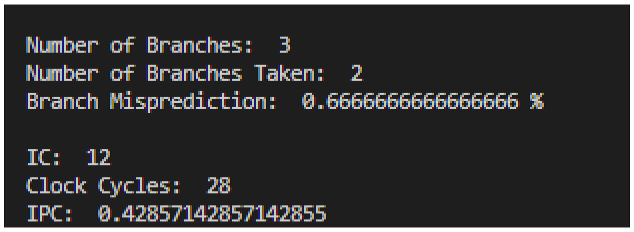


```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  27

Trace Table:
 instruction                        issue   start_exec   end_exec   write
 ['load', 'r5', '0', 'r2']              1            2          4       5
 ['load', 'r2', '1', 'r2']              2            3          5       6
 ['div', 'r0', 'r2', 'r4']              3            7         16      17
 ['add', 'r4', 'r2', 'r6']              4            7          8       9
 ['bne', 'r1', 'r2', '2']               5            7          7       8
 ['add', 'r6', 'r1', 'r5']              9           10         11      12
 ['bne', 'r6', 'r6', '5']              10           13         13      14
 ['store', 'r1', 2, 'r2']              11           15         17      19
 ['bne', 'r1', 'r2', 0]                15           16         16      18
 ['call', 6, None, None]               19           20         20      21
 ['ret', None, None, None]             22           23         23      24
 ['add', 'r0', 'r0', 'r0']             25           26         27

Resrvation Station:
 Name         Busy   OP     Vj    Vk   Qj   Qk   A   Imm
 Load1        N
 Load2        N
 Store1       N
 Store2       N
 Add/Addi_1   Y      add     0     0
 Add/Addi_2   N
 Add/Addi_3   N
 Nand         N
 Div          N
 Bne          N
 Call/Ret     N

Register Stat Table:
 r0   r1   r2   r3   r4   r5   r6   r7   r8


Register Values:
   r0   r1   r2   r3   r4   r5   r6   r7   r8
    0   12    0    3    6    0    1    7    8

Press Enter to continue or 0 to Exit:
```



```
------------- Tomasulo Algorithm Simulator -------------

Clock Cycle:  28

Trace Table:
+-------------------------------------+-------+------------+----------+-------+
| instruction                         | issue | start_exec | end_exec | write |
+-------------------------------------+-------+------------+----------+-------+
| ['load', 'r5', '0', 'r2']           |     1 |          2 |        4 |     5 |
| ['load', 'r2', '1', 'r2']           |     2 |          3 |        5 |     6 |
| ['div', 'r0', 'r2', 'r4']           |     3 |          7 |       16 |    17 |
| ['add', 'r4', 'r2', 'r6']           |     4 |          7 |        8 |     9 |
| ['bne', 'r1', 'r2', '2']            |     5 |          7 |        7 |     8 |
| ['add', 'r6', 'r1', 'r5']           |     9 |         10 |       11 |    12 |
| ['bne', 'r6', 'r6', '5']            |    10 |         13 |       13 |    14 |
| ['store', 'r1', 2, 'r2']            |    11 |         15 |       17 |    19 |
| ['bne', 'r1', 'r2', 0]              |    15 |         16 |       16 |    18 |
| ['call', 6, None, None]             |    19 |         20 |       20 |    21 |
| ['ret', None, None, None]           |    22 |         23 |       23 |    24 |
| ['add', 'r0', 'r0', 'r0']           |    25 |         26 |       27 |    28 |
+-------------------------------------+-------+------------+----------+-------+

Resrvation Station:
+-------------+------+----+----+----+----+----+---+-----+
| Name        | Busy | OP | Vj | Vk | Qj | Qk | A | Imm |
+-------------+------+----+----+----+----+----+---+-----+
| Load1       | N    |    |    |    |    |    |   |     |
| Load2       | N    |    |    |    |    |    |   |     |
| Store1      | N    |    |    |    |    |    |   |     |
| Store2      | N    |    |    |    |    |    |   |     |
| Add/Addi_1  | N    |    |    |    |    |    |   |     |
| Add/Addi_2  | N    |    |    |    |    |    |   |     |
| Add/Addi_3  | N    |    |    |    |    |    |   |     |
| Nand        | N    |    |    |    |    |    |   |     |
| Div         | N    |    |    |    |    |    |   |     |
| Bne         | N    |    |    |    |    |    |   |     |
| Call/Ret    | N    |    |    |    |    |    |   |     |
+-------------+------+----+----+----+----+----+---+-----+

Register Stat Table:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|    |    |    |    |    |    |    |    |    |
+----+----+----+----+----+----+----+----+----+

Register Values:
+----+----+----+----+----+----+----+----+----+
| r0 | r1 | r2 | r3 | r4 | r5 | r6 | r7 | r8 |
+----+----+----+----+----+----+----+----+----+
|  0 | 12 |  0 |  3 |  6 |  0 |  1 |  7 |  8 |
+----+----+----+----+----+----+----+----+----+

Press Enter to continue or 0 to Exit:
```

Results:
After you press 0 to show that the program has finished you will get a small report on your executed program.



```
Number of Branches:  3
Number of Branches Taken:  2
Branch Misprediction:  0.6666666666666666 %

IC:  12
Clock Cycles:  28
IPC:  0.42857142857142855
```

3. Results Discussion

Here, we can see that the IPC is much smaller than 1. This is because we have to stall issuing if we have a call of a return issued (this happens twice in our program), and also, if a branch is issued, we must flush our reservation station if it is taken. All this slows down the processor and reduces our IPC.

Also, notice that in the screenshot for clock cycle 28, we can clearly see that even though issuing will always happen in order, executing and writing may happen at any time as long as there are no true dependencies (RAW) or branches