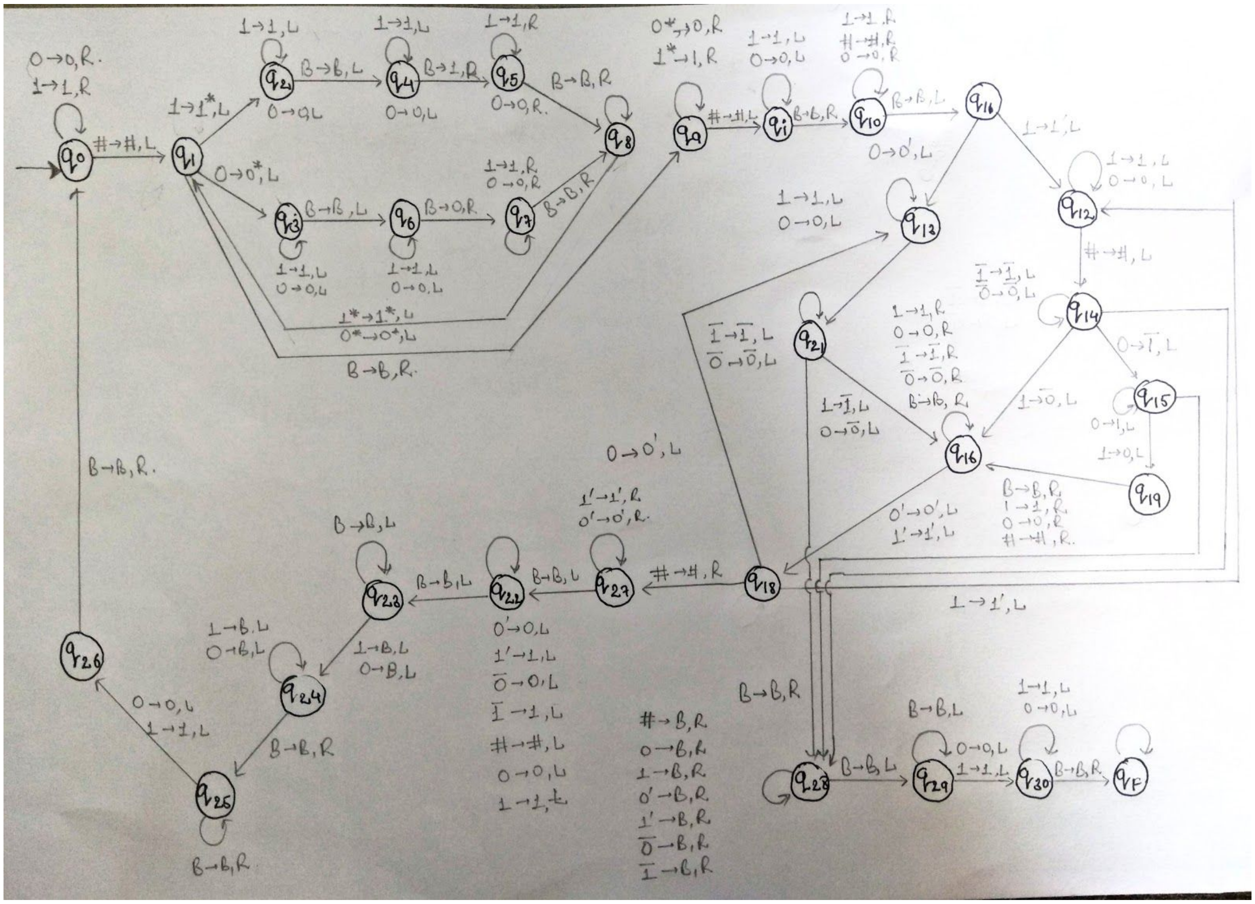
ReadMe:

- ➢ To Run of the code please follow the following instructions:
    1. Go to the folder of the code.
    2. On windows:
        a. gcc *filename.c*
        b. a
        c. Add the legible input in the program
    3. On Ubuntu:
        a. gcc -o *filename.c*
        b. ./a
        c. Add the legible input in the program

Both the questions are made on the single Turing machine model in the coding part as I have simulated the working of a simple Turing machine. In both the questions, I just have to change the configuration of the Turing Machine which includes the number of states, transition functions for every state, initial and final states. I have used 3 structures for states, transition functions, and Turing machine. Firstly I have defined the states then transition functions. Add all the transition functions to the respective state and add all the states to the Turing machine.

About Turing machines:

Question1.



In the first question, we have to find the remainder of “a”, if divided by “b”.

The basic algorithm I have approached is :

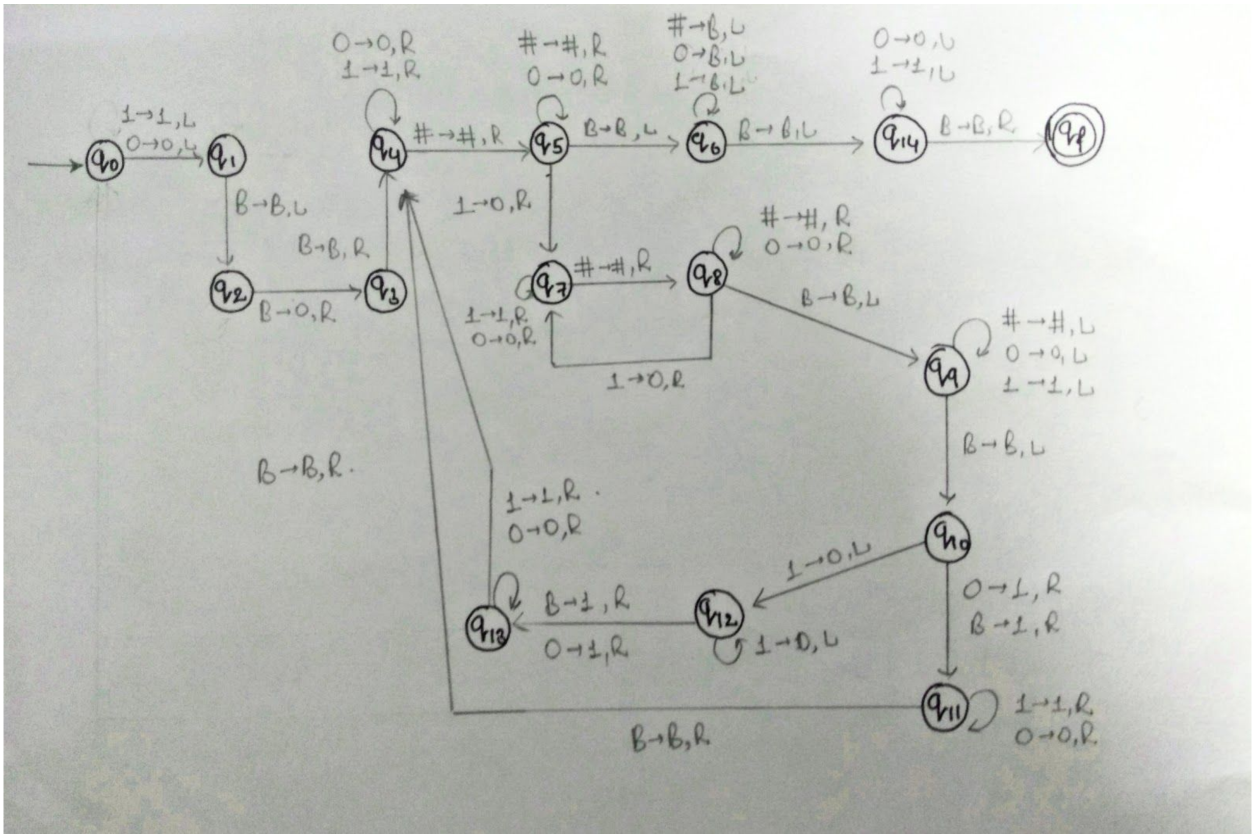
```
if(a>=b)
{
        a=a-b;
}
else
{
        a is the answer;
}
```

Now on the Turing machine, I have performed three functions i.e. copying, comparing, and subtracting.

Subtraction is in the binary form.

Note: In Coding Part I used : 1’=x,0’=y,1 bar =X,0 bar=Y,1*=A,0*=B

Question2.



In the second question, we have to find the vertex having a maximum degree.
I have assumed it as outdegree.
The basic design of the Turing machine is to find the row having the maximum number of 1 and the number of 1's in that row is the output of the answer.

I have used the binary addition whenever I find a 1 in one complete iteration over the input.
If I get one 1 in one row ignore all zeroes and ones and change that 1 to 0.Perform binary addition at the end of the iteration and whenever I find no 1 in all the rows answer is the added value.

Attaching the Scheme of both the Turing machines which I verified with the online Turing machine.
Link-https://turingmachinesimulator.com/

My Code is also somewhat motivated by this platform.

If my code got stuck you can verify the design of the Turing machine by just copy-pasting the attached scheme of respective questions.