Lab: Implement the TM accepting the language { 0^n1^n / n ≥ 1} over alphabet, $\Sigma = \{0, 1\}$.

Turing Machine Description:

Given finite sequence of 0's and 1's on tape and followed by blanks. The TM starts at state q0 and changes 0 to an X and moves to the right changing its state to q1.

At state q1, TM expects 1 and changes a 1 to Y and moves to the left changing the state to q2. If any number of 0s and Ys are seen, it remains on the state q1 and leaving these symbols unchanged and moving the head position to the right.

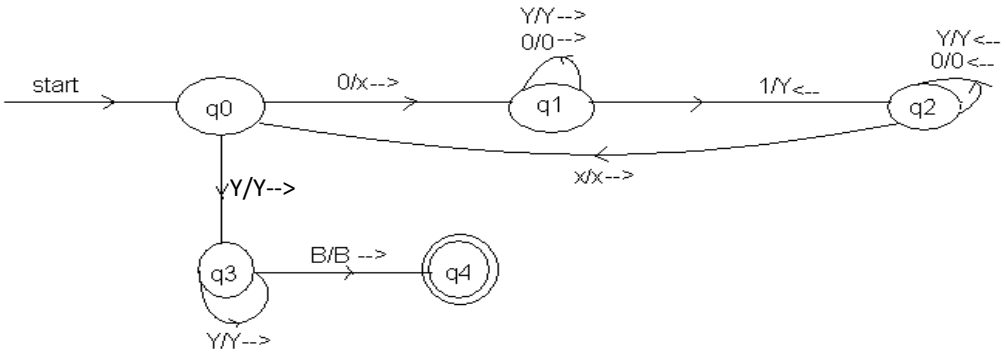
At state q2, if 0s or Y's are seen, it leaves them as it is and moves to the left staying at the same state q2. If it sees X at state q2, the tape symbol is left unchanged moves to right switching its state to q0.

At state q0, if it sees Y then the symbol is left unchanged and head is moved right changing to the state q3.

At state q3, if Y is seen, it is left unchanged and head is moved to the right. If BLANK (here, '\0') is seen at state q3, the string is accepted by switching the state to q4.

At any state, if the machine seen other than the defined symbols, it rejects the string.

The state transition diagram is shown in the figure below.



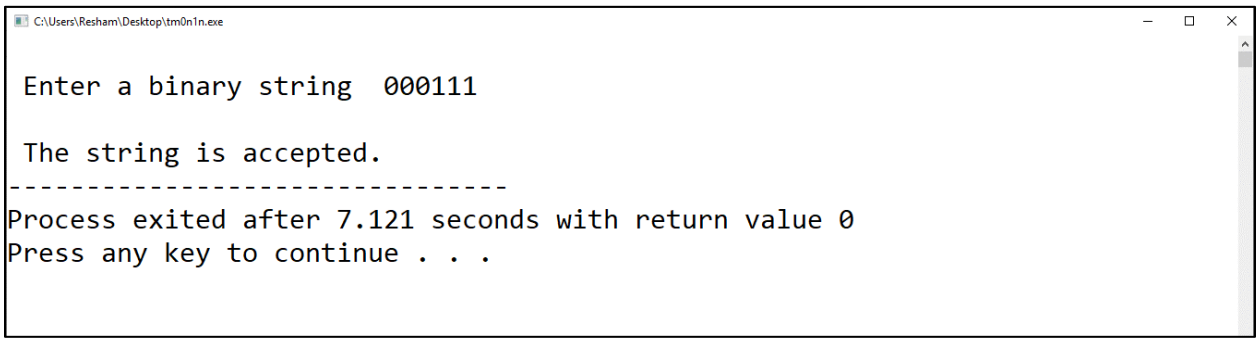
Code:

```
/* TOC Lab: Implement a TM for L = { set of all strings over {0,1} such that the string have number of 0s followed by same number of 1s. */
#include<stdio.h>

enum states { q0, q1, q2,q3,q4,qr};

int main()
{
        char input[100];
        enum states curr_state = q0;
```

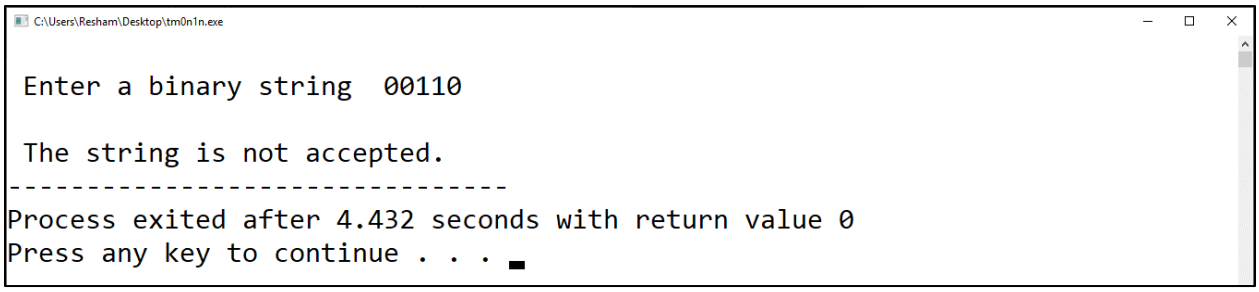
```
int i;
for(i=0;i<100;i++)
   input[i] = '\0';
printf("\n Enter a binary string\t");
gets(input);
i=0;
while(1)
{
        switch(curr_state)
        {
                case q0:
                        if(input[i]=='0')
                        {
                                curr_state = q1;
                                input[i]='x';
                                i++;
                        }
                        else if(input[i]=='y')
                        {
                                curr_state=q3;
                                i++;
                        }
                        else
                                curr_state = qr;  // for invalid transition
                        break;
                case q1:
                        if(input[i]=='0')
                        {
                                curr_state = q1;
                                i++;
                        }
                        else if(input[i]=='y')
                        {
                                curr_state = q1;
                                i++;
                        }
                        else if(input[i]=='1')
                        {
                                curr_state = q2;
                                input[i]='y';
                                i--;
                        }
                        else
                                  curr_state = qr;
                        break;
                case q2:
                        if(input[i]=='0')
```

```
                                        {
                                                curr_state = q2;
                                                i--;
                                        }
                                        else if(input[i]=='y')
                                        {
                                                curr_state = q2;
                                                i--;
                                        }
                                        else if(input[i]=='x')
                                        {
                                                curr_state = q0;
                                                i++;
                                        }
                                        else
                                                curr_state = qr;
                                        break;
                                case q3:
                                        if(input[i]=='y')
                                        {
                                                curr_state = q3;
                                                i++;
                                        }
                                        else if(input[i]=='\0')
                                        {
                                                curr_state=q4;
                                        }
                                        else
                                                curr_state = qr;
                                        break;
                        }//end of switch
                         if(curr_state == qr || curr_state==q4)
                            break;
                }//end of while loop

 if(curr_state == q4)
                printf("\n The string is accepted.");
 else
                printf("\n The string is not accepted.");

   return 0;
}
```

OUTPUT


C:\Users\Resham\Desktop\tm0n1n.exe
Enter a binary string 000111
The string is accepted.

Process exited after 7.121 seconds with return value 0
Press any key to continue . . .


C:\Users\Resham\Desktop\tm0n1n.exe
Enter a binary string 00110
The string is not accepted.

Process exited after 4.432 seconds with return value 0
Press any key to continue . . .