

Experiment 1

Design a Finite State Machine (FSM) that accepts all strings over input symbols $\{0, 1\}$ having three consecutive 1's as a substring.

Discussion:

As per the AIM, set of valid strings are represented by set A:

$A = \{111, 0111, 1110, 0101011110101, \dots\}$

means any string should be declared valid if it contains 111 as a substring. Let M be the machine for above AIM, hence it can be defined as $M(Q, \Sigma, \delta, q_0, F)$ where

Q: set of states: $\{A, B, C, D\}$

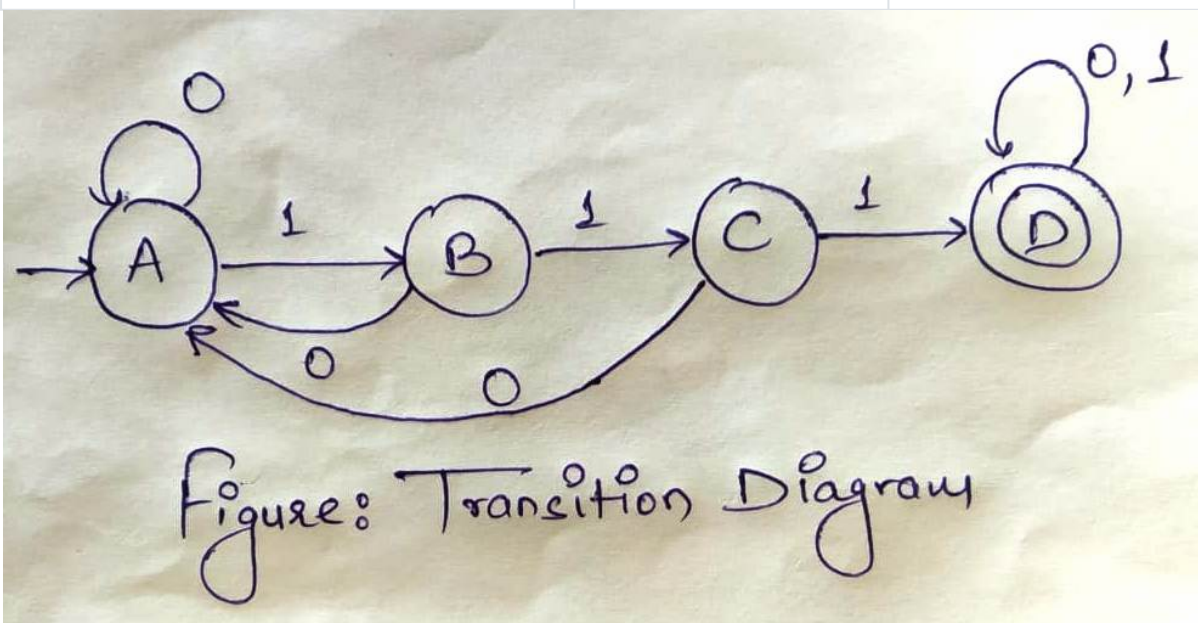
Σ : set of input symbols: $\{0, 1\}$

q_0 : initial state (A)

F: set of Final states: $\{D\}$

δ : Transition Function: (Transition state diagram is shown in Figure 1.)

State	Input	
-	0	1
A	A	B
B	A	C
C	A	D
D	D	D



```

#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void main()
{
    char Input[100];
    clrscr();
    cout<<"Enter a string to validate (input string should be of 0 and 1)\n";
    gets(Input);
    int i=-1;
    A:
        i++;
        if(Input[i]=='0')
        {
            goto A;
        }
        else if(Input[i]=='1')
        {
            goto B;
        }
        else if(Input[i]=='\0')
        {
            goto Invalid;
        }
        else
        {
            goto Wrong;
        }
    B:
        i++;
        if(Input[i]=='0')

```

```
{  
    goto A;  
}  
else if(Input[i]=='1')  
{  
    goto C;  
}  
else if(Input[i]=='\0')  
{  
    goto Invalid;  
}  
else  
{  
    goto Wrong;  
}
```

C:

```
i++;  
if(Input[i]=='0')  
{  
    goto A;  
}  
else if(Input[i]=='1')  
{  
    goto D;  
}  
else if(Input[i]=='\0')  
{  
    goto Invalid;  
}  
else  
{  
    goto Wrong;
```

```

    }

D:

    i++;
    if(Input[i]=='0')
    {
        goto D;
    }
    else if(Input[i]=='1')
    {
        goto D;
    }
    else if(Input[i]=='\0')
    {
        goto Valid;
    }
    else
    {
        goto Wrong;
    }

Valid:

    cout<<"\n Output: Valid String";
    goto exit;

Invalid:

    cout<<"\n Output: Invalid String";
    goto exit;

Wrong:

    cout<<"\n Please enter binary string {format of 0, 1}";

exit:

    getch();
}

```

Experiment 2

Design a Program for creating machine that accepts the string always ending with 10 or 11

```
#include <bits/stdc++.h>

using namespace std;

// Various states of DFA machine are defined
// using functions.
void q1(string, int);
void q2(string, int);
void q3(string, int);
void q4(string, int);

// End position is checked using the string
// length value.
// q0 is the starting state.
// q1 and q2 are intermediate states.
// q3 and q4 are final states.
void q1(string s, int i)
{
    cout << "q1->";

    if (i == s.length()) {
        cout << "NO \n";
        return;
    }

    // state transitions
    // 0 takes to q1, 1 takes to q3
    if (s[i] == '0')
        q1(s, i + 1);
    else
        q3(s, i + 1);
}
```

```
void q2(string s, int i)
```

```
{
```

```
    cout << "q2->";
```

```
    if (i == s.length()) {
```

```
        cout << "NO \n";
```

```
        return;
```

```
}
```

```
// state transitions
```

```
// 0 takes to q4, 1 takes to q2
```

```
if (s[i] == '0')
```

```
    q4(s, i + 1);
```

```
else
```

```
    q2(s, i + 1);
```

```
}
```

```
void q3(string s, int i)
```

```
{
```

```
    cout << "q3->";
```

```
    if (i == s.length()) {
```

```
        cout << "YES \n";
```

```
        return;
```

```
}
```

```
// state transitions
```

```
// 0 takes to q4, 1 takes to q2
```

```
if (s[i] == '0')
```

```
    q4(s, i + 1);
```

```
else
```

```
    q2(s, i + 1);
```

```
}
```

```
void q4(string s, int i)
```

```
{  
    cout << "q4->";  
    if (i == s.length()) {  
        cout << "YES \n";  
        return;  
    }  
  
    // state transitions  
    // 0 takes to q1, 1 takes to q3  
    if (s[i] == '0')  
        q1(s, i + 1);  
    else  
        q3(s, i + 1);  
}
```

```
void q0(string s, int i)
```

```
{  
    cout << "q0->";  
    if (i == s.length()) {  
        cout << "NO \n";  
        return;  
    }  
}
```

```
    // state transitions  
    // 0 takes to q1, 1 takes to q2  
    if (s[i] == '0')  
        q1(s, i + 1);  
    else  
        q2(s, i + 1);  
}
```

```
// Driver Code
```

```
int main()
```

```
{
```

```

string s = "010101";

// all state transitions are printed.

// if string is accpetable, YES is printed.

// else NO is printed

cout << "State transitions are ";

q0(s, 0);

}

```

Experiment 3

Design a Finite State Machine (FSM) that accepts all strings over input symbols {0, 1} mean binary number which are divisible by 3.

As per the AIM, set of valid strings are represented by set A:

$A = \{0, 00, 000, 11, 011, 110, \dots\}$

means any binary string that when divide by three gives remainder zero. Let M be the machine for above AIM, hence it can be define as $M(Q, \Sigma, \delta, q_0, F)$ where

Q: set of states: {q, q0, q1, q2}

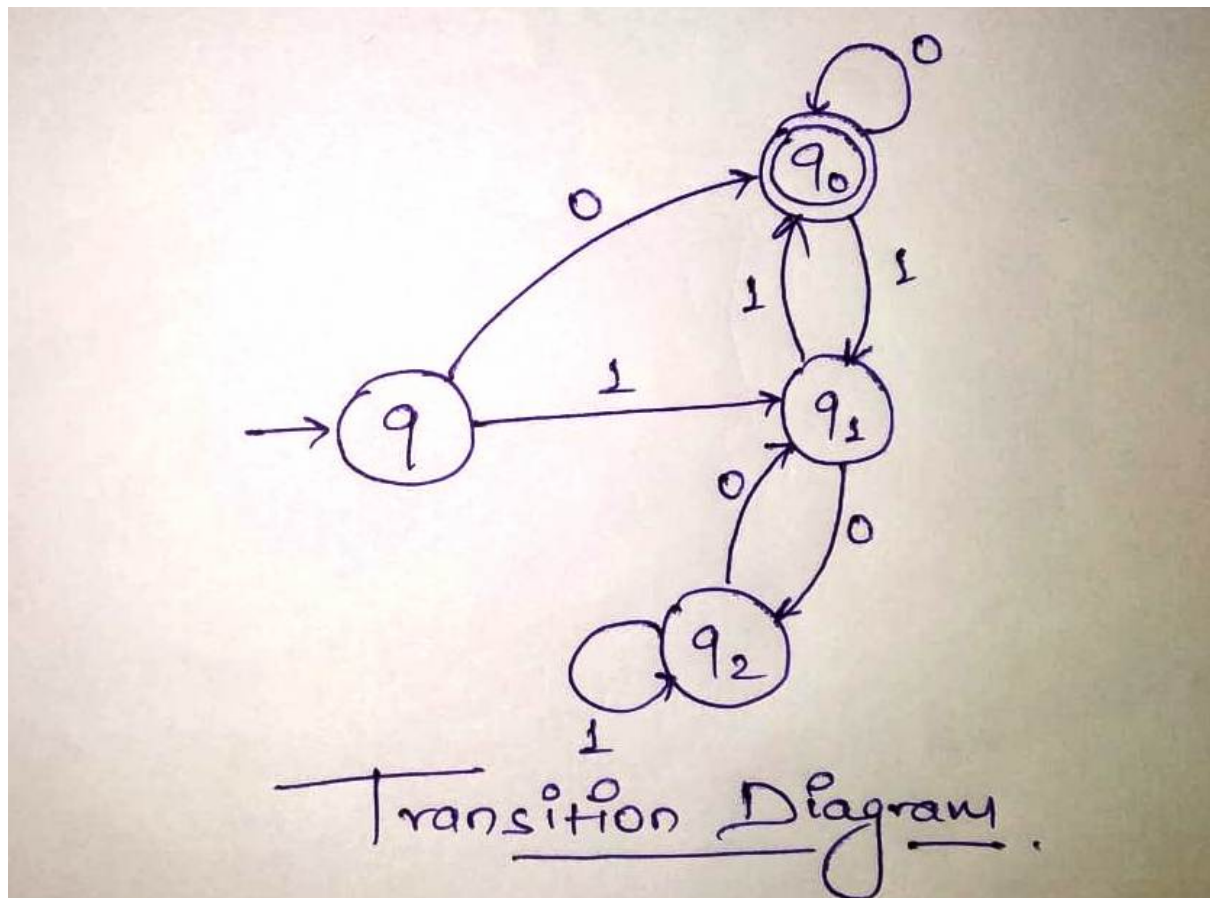
Σ : set of input symbols: {0, 1}

q0: initial state (q)

F: set of Final states: {q0}

δ : Transition Function: (Transition state diagram is shown in Figure 1.)

State	Input	
-	0	1
q	q0	q1
q0	q0	q1
q1	q2	q0
q2	q1	q2



```

#include <iostream.h>
#include <conio.h>
#include <stdio.h>
void main()
{
    char Input[100];
    clrscr();
    cout<<"Enter a string to validate (input string should be of 0 and 1)\n";
    gets(Input);
    int i=-1;
    q:
        i++;
        if(Input[i]=='0')
        {
            goto q0;
        }
    }

```

```

    }
    else if(Input[i]=='1')
    {
        goto q1;
    }
    else if(Input[i]=='\0')
    {
        goto Invalid;
    }
    else
    {
        goto Wrong;
    }
q0:
    i++;
    if(Input[i]=='0')
    {
        goto q0;
    }
    else if(Input[i]=='1')
    {
        goto q1;
    }
    else if(Input[i]=='\0')
    {
        goto Valid;
    }
    else
    {
        goto Wrong;
    }
q1:

```

```
    i++;  
    if(Input[i]=='0')  
    {  
        goto q2;  
    }  
    else if(Input[i]=='1')  
    {  
        goto q0;  
    }  
    else if(Input[i]=='\0')  
    {  
        goto Invalid;  
    }  
    else  
    {  
        goto Wrong;  
    }  
q2:  
    i++;  
    if(Input[i]=='0')  
    {  
        goto q1;  
    }  
    else if(Input[i]=='1')  
    {  
        goto q2;  
    }  
    else if(Input[i]=='\0')  
    {  
        goto Invalid;  
    }  
    else
```

```

        {
            goto Wrong;
        }
Valid:
    cout<<"\n Output: Valid String";
    goto exit;
Invalid:
    cout<<"\n Output: Invalid String";
    goto exit;
Wrong:
    cout<<"\n Please enter binary string {format of 0, 1}";
exit:
    getch();
}

```

Experiment 4

Design a Finite State Machine (FSM) that accepts all decimal string which are divisible by 3.

As per the AIM, set of valid strings are represented by set A:

$A = \{0, 3, 6, 9, 03, 06, 09, 12, 012, \dots\}$

means any decimal number string that when divided by three gives remainder zero. Let M be the machine for above AIM, hence it can be define as $M(Q, \Sigma, \delta, q_0, F)$ where

Q: set of states: $\{q, q_0, q_1, q_2\}$

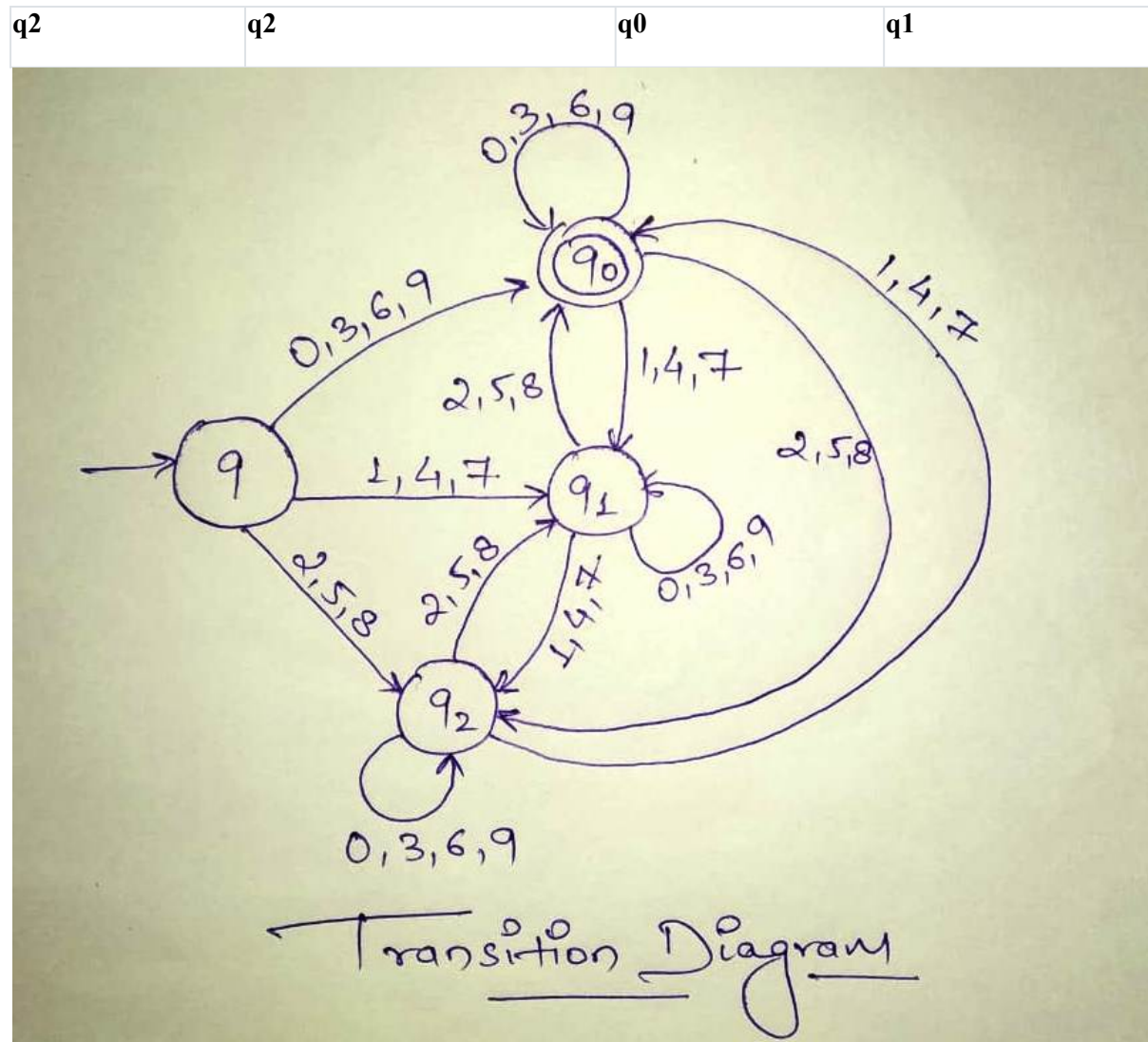
Σ : set of input symbols: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

q_0 : initial state (q)

F: set of Final states: $\{q_0\}$

δ : Transition Function: (Transition state diagram is shown in Figure 1.)

State	Input		
-	0, 3, 6, 9	1, 4, 7	2, 5, 8
q	q0	q1	q2
q0	q0	q1	q2
q1	q1	q2	q0



```

#include <iostream.h>
#include <conio.h>
#include <stdio.h>

void main()
{
    char Input[100];
    clrscr();

    cout<<"Enter a string to validate (input string should be decimal number (i.e
constructed from 0,1,2,3,4,5,6,7,8,9 digits)\n";

    gets(Input);

    int i=-1;

    q:
        i++;

```

```

    if(Input[i]=='0' || Input[i]=='3' || Input[i]=='6' || Input[i]=='9')
    {
        goto q0;
    }
    else if(Input[i]=='1' || Input[i]=='4' || Input[i]=='7')
    {
        goto q1;
    }
    else if(Input[i]=='2' || Input[i]=='5' || Input[i]=='8')
    {
        goto q2;
    }
    else if(Input[i]=='\0')
    {
        goto Invalid;
    }
    else
    {
        goto Wrong;
    }
q0:
    i++;
    if(Input[i]=='0' || Input[i]=='3' || Input[i]=='6' || Input[i]=='9')
    {
        goto q0;
    }
    else if(Input[i]=='1' || Input[i]=='4' || Input[i]=='7')
    {
        goto q1;
    }
    else if(Input[i]=='2' || Input[i]=='5' || Input[i]=='8')
    {

```

```

        goto q2;
    }
    else if(Input[i]=='\0')
    {
        goto Valid;
    }
    else
    {
        goto Wrong;
    }
q1:
    i++;
    if(Input[i]=='0' || Input[i]=='3' || Input[i]=='6' || Input[i]=='9')
    {
        goto q1;
    }
    else if(Input[i]=='1' || Input[i]=='4' || Input[i]=='7')
    {
        goto q2;
    }
    else if(Input[i]=='2' || Input[i]=='5' || Input[i]=='8')
    {
        goto q0;
    }
    else if(Input[i]=='\0')
    {
        goto Invalid;
    }
    else
    {
        goto Wrong;
    }

```

q2:

```
i++;  
if(Input[i]=='0' || Input[i]=='3' || Input[i]=='6' || Input[i]=='9')  
{  
    goto q2;  
}  
else if(Input[i]=='1' || Input[i]=='4' || Input[i]=='7')  
{  
    goto q0;  
}  
else if(Input[i]=='2' || Input[i]=='5' || Input[i]=='8')  
{  
    goto q1;  
}  
else if(Input[i]=='\0')  
{  
    goto Invalid;  
}  
else  
{  
    goto Wrong;  
}
```

Valid:

```
cout<<"\n Output: Valid String";  
goto exit;
```

Invalid:

```
cout<<"\n Output: Invalid String";  
goto exit;
```

Wrong:

```
cout<<"\n Please enter valid decimal number string";
```

exit:

```
getch();
```


}