Compiler Design
(Assignment-1)

Question1:Implementation of Language recognizer for set of all strings over input alphabet ∑={a,b} containing an even number of a's and even number of b's.

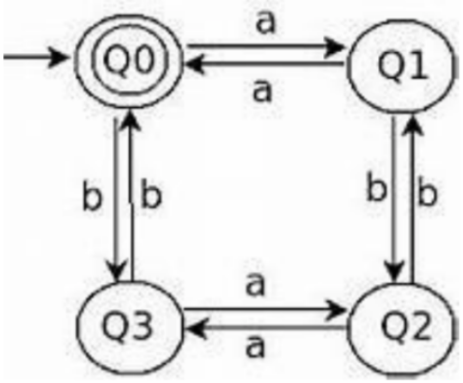
Description:

The acceptable strings of the language are ε(Null string), aa, bb, abba, babbab etc.

Deterministic Finite Automata for the given language is given below:

DFA M=(Q,∑,δ,Q_0,F) Where
Q=Set of all states ={Q_0,Q_1,Q_2,Q_3}
∑=Input Alphabet={a,b},
Start state is Q_0
F=Set of all final States={ Q_0}

And the transitions are defined in the transition diagram



Algorithm:Language recognizer
Input:
input //input string
Output:
Algorithm prints a message
"String accepted": If the input is acceptable by the language,

“String not accepted” otherwise,
“Invalid token”: If the input string contains symbols other than input alphabet.

Method:

```
state=0 //initial state
while((current=input[i++])!='\0'){
switch(state)
case 0: if(current=='a') state=1;
else if(current=='b') state=2;
else
Print "Invalid token" ; exit;
case 1: if(current=='a') state=0;
else if(current=='b') state=3;
else
Print "Invalid token" ; exit;
case 2: if(current=='a') state=3;
else if(current=='b') state=0;
else
Print "Invalid token" ; exit;
case 3: if(current=='a') state=2;
else if(current=='b') state=1;
else
Print "Invalid token" ; exit;
end switch
end while
//Print output
if(state==0)
Print ”String accepted”
else
Print ”String not accepted”
```

C Code:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int state=0,i=0; char current,input[20];
printf("Enter input string \t :");
scanf("%s",input);
while((current=input[i++])!='\0')
{
switch(state)
 {
case 0:
  if(current=='a')
      state=1;
else if(current=='b')
                state=2;
    else
    {
   printf("Invalid token");
    exit(0);
    }
    break;

case 1:
if(current=='a')
   state=0;
else if(current=='b')
              state=3;
else
{
printf("Invalid token");
 exit(0);
```

```
 }
break;

case 2:
 if(current=='a')
        state=3;
  else if(current=='b')
  state=0;
  else
  {
     printf("Invalid token");
     exit(0);
  }
  break;
case 3:
     if(current=='a')
            state=2;
    else if(current=='b')
         state=1;
    else
    {
     printf("Invalid token");
     exit(0);
    }
    break;
}
}
//end switch end while
//Print output
if(state==0)
  printf("String accepted");
else
  printf ("String not accepted");
```

}

Test Cases:

Test Case	Input String	Output
1)	aabb	String Accepted
2)	abab	String Accepted
3)	aaabb	String not Accepted
4)	abcd	Invalid Token

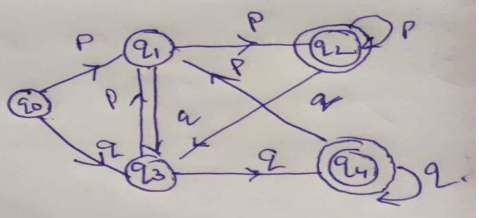
Question 2:Implementation of Language recognizer for a set of all strings ending with two symbols of the same type.

Description:

The acceptable strings of the Language are PPqq, Pqq, qPPqq etc.

Deterministic Finite Automata for the given Language is given below.

DFA M = (Q, Σ, δ, Q0, F) where
Q = set of all states {q0, q1, q2, q3, q4}
Σ = Input alphabets = {p, q}
State (initial) is 'q0'.
F = Set of all final states = {q2, q4}



Algorithm:Language recognizer
Input:
input //input string
Output:
Algorithm prints a message
“String accepted”: If the input is acceptable by the language,
“String not accepted” otherwise,
“Invalid token”: If the input string contains symbols other than the input alphabet.

Method:

```
state=1 //initial state
for(i=0; (current=c[i])!='\0'; i++){
switch(state)
case 1: if(current=='p') state=2;
else if(current=='q') state=4;
else
Print "Invalid token" ; exit;
case 2: if(current=='p') state=3;
else if(current=='q') state=4;
else
Print "Invalid token" ; exit;
case 3: if(current=='p') state=3;
else if(current=='4') state=4;
else
Print "Invalid token" ; exit;
case 4: if(current=='p') state=2;
else if(current=='q') state=5;
else
Print "Invalid token" ; exit;
case 5: if(current=='p') state=2;
else if(current=='q') state=5;
```

```
else
Print "Invalid token" ; exit;

end switch
end while
//Print output
if(state==3 or state==5)
Print ”String accepted”
else
Print ”String not accepted”
```

C Code:

```
#include <stdio.h>
#include<stdlib.h>
int main()
{
 //Assuming initial state Q0 as state 1
  int state=1,i=0;
  //Taking a character array of size 30
  char c[30];
  //Taking a character ‘current’ to check the input string
  char current;
  printf("Enter your String:");
  scanf("%s",c);
       for(i=0; (current=c[i])!='\0'; i++)
        {
          switch(state)
          {
              //Taking my input alphabets as 'P' and 'Q'
              case 1: if(current=='p')  //When current state is at Q0 in DFA
                      {
```

```
        state=2;
      }
     else if(current=='q')
          state=4;
     else
       {
        printf("Invalid token");
        exit(0);
       }
       break;

case 2: if(current=='p') //When current state is at Q1 in DFA
          state=3;
     else if(current=='q')
          state=4;
     else
       {
        printf("Invalid token");
        exit(0);
       }
       break;

case 3: if(current=='p') //When current state is at Q2 in DFA
          state=3;
     else if(current=='q')
          state=4;
     else
      {
        printf("Invalid token");
        exit(0);
      }
      break;
case 4: if(current=='p') //When current state is at Q3 in DFA
```

```
                    state=2;
                else if(current=='q')
                    state=5;
                else
                  {
                   printf("Invalid token");
                   exit(0);
                  }
                  break;
            case 5: if(current=='p') //When current state is at Q4 in DFA
                    state=2;
                else if(current=='q')
                    state=5;
                else
                  {
                   printf("Invalid token");
                   exit(0);
                  }
                  break;
        }
      }
  //If the current state is at either Q2 or Q4,that String is accepted
  //not accepted.
  if(state==3 || state==5)
  {
    printf("String is Accepted");
  }
  else
   {
    printf("String is not Accepted");
   }
}
```

Test Cases:

Test Case	Input String	Output
1)	ppqq	String is accepted
2)	qqpqx	Invalid Token
3)	pqpqpqqppqp	String is not accepted
4)	qppqqppqpqpqpqppqq q	String is accepted