EXP 2

CONVERSION OF REGULAR EXPRESSION TO NFA

AIM: To write a program for converting Regular Expression to NFA.

PROCEDURE:

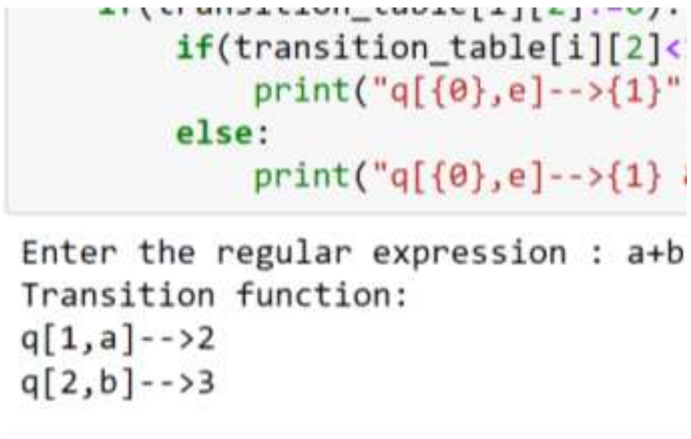
- Start
- Get the input from the user
- Initialize separate variables & functions for Postfix , Display NFA
- Create separate methods for different operators like +,*, .
- By using Switch case Initialize different cases for the input
- For ' . ' operator Initialize a separate method by using various stack functions do the same for the other operators like ' * ' and ' + '.
- Regular expression is in the form like a.b (or) a+b
- 8. Display the output
- 9. Stop

CODE:

```
transition_table = [ [0]*3 for _ in range(20) ]


re = input("Enter the regular expression : ")

re += " "


i = 0

j = 1

while(i<len(re)):

    if re[i] == 'a':

        try:

            if re[i+1] != '|' and re[i+1] !='*':

                transition_table[j][0] = j+1
            j += 1
        elif re[i+1] == '|' and re[i+2] =='b':
            transition_table[j][2]=((j+1)*10)+(j+3)
            j+=1
            transition_table[j][0]=j+1
            j+=1
            transition_table[j][2]=j+3
            j+=1
            transition_table[j][1]=j+1
            j+=1
            transition_table[j][2]=j+1
            j+=1
            i=i+2
        elif re[i+1]=='*':
            transition_table[j][2]=((j+1)*10)+(j+3)
            j+=1
            transition_table[j][0]=j+1
            j+=1
            transition_table[j][2]=((j+1)*10)+(j-1)
            j+=1
    except:
        transition_table[j][0] = j+1

elif re[i] == 'b':
    try:
        if re[i+1] != '|' and re[i+1] !='*':
            transition_table[j][1] = j+1
            j += 1
        elif re[i+1]=='|' and re[i+2]=='a':
            transition_table[j][2]=((j+1)*10)+(j+3)
            j+=1
            transition_table[j][1]=j+1
            j+=1
            transition_table[j][2]=j+3
            j+=1
            transition_table[j][0]=j+1
            j+=1
            transition_table[j][2]=j+1
            j+=1
            i=i+2
        elif re[i+1]=='*':
            transition_table[j][2]=((j+1)*10)+(j+3)
            j+=1
            transition_table[j][1]=j+1
            j+=1
            transition_table[j][2]=((j+1)*10)+(j-1)
            j+=1
    except:
        transition_table[j][1] = j+1

elif re[i]=='e' and re[i+1]!='|'and re[i+1]!='*':
    transition_table[j][2]=j+1
```

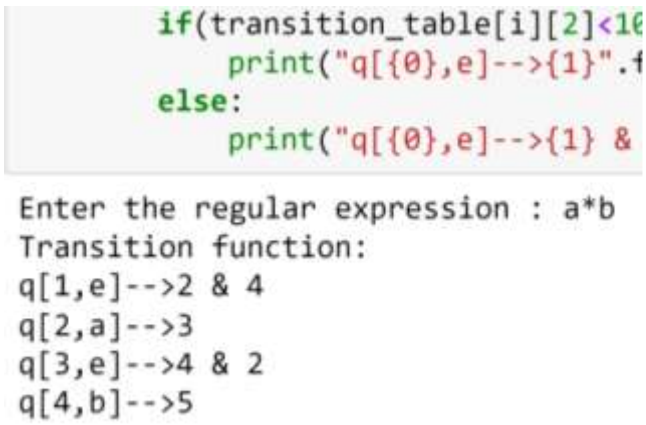
```
        j+=1

    elif re[i]==')' and re[i+1]=='*':

        transition_table[0][2]=((j+1)*10)+1
        transition_table[j][2]=((j+1)*10)+1
        j+=1

    i +=1

print ("Transition function:")
for i in range(j):
    if(transition_table[i][0]!=0):
        print("q[{0},a]-->{1}".format(i,transition_table[i][0]))
    if(transition_table[i][1]!=0):
        print("q[{0},b]-->{1}".format(i,transition_table[i][1]))
    if(transition_table[i][2]!=0):
        if(transition_table[i][2]<10):
            print("q[{0},e]-->{1}".format(i,transition_table[i][2]))
        else:
            print("q[{0},e]-->{1} &
{2}".format(i,int(transition_table[i][2]/10),transition_table[i][2]%10))
```

OUTPUT:



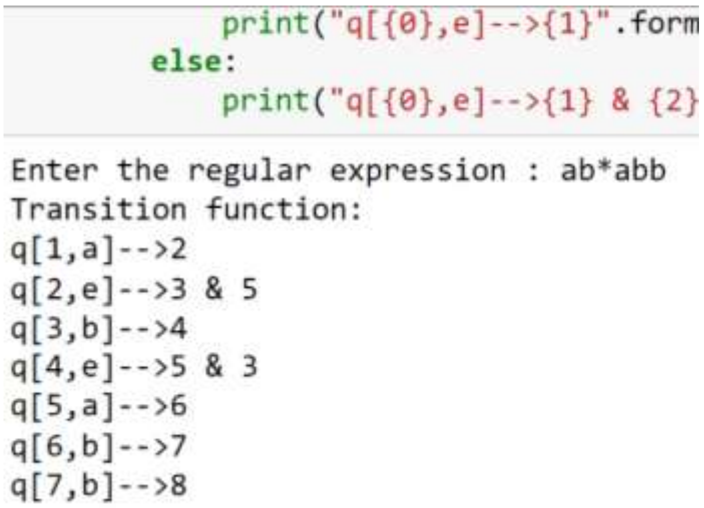
```
        if(transition_table[i][2]<
            print("q[{0},e]-->{1}"
        else:
            print("q[{0},e]-->{1}
```

```
Enter the regular expression : a+b
Transition function:
q[1,a]-->2
q[2,b]-->3
```



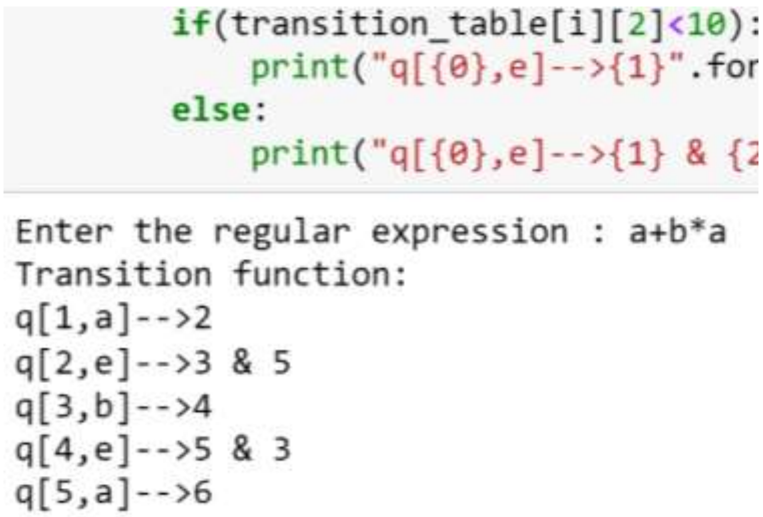
```
        if(transition_table[i][2]<1
            print("q[{0},e]-->{1}".
        else:
            print("q[{0},e]-->{1} &
```

```
Enter the regular expression : a*b
Transition function:
q[1,e]-->2 & 4
q[2,a]-->3
q[3,e]-->4 & 2
q[4,b]-->5
```



```
            print("q[{0},e]-->{1}".form
        else:
            print("q[{0},e]-->{1} & {2}
```

```
Enter the regular expression : ab*abb
Transition function:
q[1,a]-->2
q[2,e]-->3 & 5
q[3,b]-->4
q[4,e]-->5 & 3
q[5,a]-->6
q[6,b]-->7
q[7,b]-->8
```



```
        if(transition_table[i][2]<10):
            print("q[{0},e]-->{1}".for
        else:
            print("q[{0},e]-->{1} & {2
```

```
Enter the regular expression : a+b*a
Transition function:
q[1,a]-->2
q[2,e]-->3 & 5
q[3,b]-->4
q[4,e]-->5 & 3
q[5,a]-->6
```

<u>RESULT:</u>

The program to convert regular expressions to NFA was implemented successfully

RA1911003010387

CHITRALEKHA.CH