

## Program: 05

### Problem Statement:

Write a program to simulate the behavior of DFA which accepts the string in which substring is 'aba'.

### Solution:

---

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
int main(){
    char s[100];
    int i=0, state=0;
    clrscr();
    printf("Enter any string:");
    gets(s);
    while(s[i]!='\0')
    {
        switch(state)
        {
            case 0:
            {
                if(s[i]=='b')
                {
                    state=0;
                }
                else if(s[i]=='a')
                {
                    state=1;
                }
                else
                {
                    printf("Error");
                    return 0;
                }
                break;
            }
            case 1:
            {
                if(s[i]=='a')
                {
                    state=1;
                }
                else if(s[i]=='b')
                {
                    state=2;
                }
            }
        }
    }
}
```

```

        }
        else
        {
            printf("Error");
            return 0;
        }
        break;
    }
    case 2:
    {
        if(s[i]=='a')
        {
            state=3;
        }
        else if(s[i]=='b')
        {
            state=0;
        }
        else
        {
            printf("Error");
            return 0;
        }
        break;
    }
    case 3:
    {
        if(s[i]=='a')
        {
            state=3;
        }
        else if(s[i]=='b')
        {
            state=3;
        }
        else
        {
            printf("Error");
            return 0;
        }
        break;
    }
}
i++;
}
if(state==3)
{
    printf("\nString Accepted");
}
else
{
    printf("\nString Not Accepted");
}

```

```
        getch();  
        return 0;  
}
```

### **Output:**

Enter any string:abb

String Not Accepted

Enter any string:abbabb

String Not Accepted

Enter any string:babbaba

String Accepted

## Program: 06

### **Problem Statement:**

Write a program to implement minimal DFA that accepts all strings of 'a' and 'b' where string contain even number of 'b' and odd number of 'a'.

### **Solution:**

---

```
#include <iostream.h>

using namespace std;

int main()
{
    string s;
    cout<<"Enter String:"<<endl;
    cin>>s;
    int ca=0,cb=0;
    for(int i=0;i<s.size();i++){
        if(s[i]=='a'){
            ca++;
        }
        else if(s[i]=='b'){
            cb++;
        }
        else{
            printf("Error");
            return 0;
        }
    }
    if(ca%2==1&&cb%2==0){
        cout<<"String accepted"<<endl;
    }
    else{
        cout<<"String not accepted"<<endl;
    }
    return 0;
}
```

**Output:**

Enter String:

ab

String not accepted

Enter String:

abbabbbab

String accepted

## Program: 07

### **Problem Statement:**

Write a program to design a DFA that accepts all strings of 'a' and 'b' where string contain exactly two a's.

### **Solution:**

---

```
#include <iostream.h>

using namespace std;

int main()
{
    string s;
    cout<<"Enter String:"<<endl;
    cin>>s;
    int ca=0;
    for(int i=0;i<s.size();i++){
        if(s[i]=='a'){
            ca++;
            if(ca>2){
                cout<<"String not accepted";
                return 0;
            }
        }
        else if(s[i]!='b'){
            cout<<"Error";
        }
    }
    if(ca==2){
        cout<<"String accepted"<<endl;
    }
    else{
        cout<<"String NOT accepted"<<endl;
    }
}
```

**Output:**

Enter String:

abb

String NOT accepted

Enter String:

ababa

String not accepted

Enter String:

babba

String accepted

## Program: 08

### **Problem Statement:**

Write a program to implement minimal DFA that accepts all strings of 'a' and 'b' where string neither ends with 'ab' nor with 'ba'.

### **Solution:**

---

```
#include <iostream>

using namespace std;

int main()
{
    string s;
    int i=0,state=0;
    cout<<"Enter String:"<<endl;
    cin>>s;
    for(int i=0;i<s.size();i++){
        if(s[i]=='a' && state == 0){
            state=1;
        }
        else if(s[i]=='b' && state == 0){
            state=3;
        }
        else if(s[i]=='a' && state == 1){
            state=1;
        }
        else if(s[i]=='b' && state == 1){
            state=2;
        }
        else if(s[i]=='b' && state == 3){
            state=3;
        }
    }
}
```



```

    }
    else if(s[i]=='a' && state == 3){
        state=2;
    }
    else if(s[i]=='b' || s[i]=='a' && state == 2){
        state=2;
    }
    else{
        printf("Error");
        return 0;
    }
}
if(state!=2){
    cout<<"String accepted"<<endl;
}
else{
    cout<<"String not accepted"<<endl;
}
return 0;
}

```

**Output:**

Enter String: bbb

String accepted

Enter String:

aa

String accepted

Enter String:

abb

String not accepted

## Program: 09

### **Problem Statement:**

Design a Program to create PDA machine that accept the well-formed parenthesis.

### **Solution:**

---

```
#include <iostream>
#include <stack>
#include <cstring>
using namespace std;
int main() {
    string input;
    stack<char> paren;
    paren.push('z');
    cout<<"Enter any String:";
    cin>>input;

    for(int i=0;i<=input.size();i++){
        if(input[i]=='(' || input[i]=='[' || input[i]=='{'){
            paren.push(input[i]);
        }

        else {

            if(!paren.empty()){

                if(input[i]==')' && paren.top()=='(' || input[i]==']' && paren.top()=='[' || input[i]=='}'
&& paren.top()=='{'){

                    paren.pop();

                }

            }

            else{

                cout<<"Invalid Paranthesis"<<endl;

                return 0;

            }

        }

    }

}
```

```
}
```

```
if(paren.top()=='z'){  
    cout<<"Paranthesis Accepted"<<endl;  
}  
else{  
    cout<<"Invalid Paranthesis"<<endl;  
}  
return 0;  
}
```

**Output:**

Enter any String: {[()]}

Paranthesis Accepted

Enter any String: {[}]

Invalid Paranthesis

## Program: 10

### **Problem Statement:**

Write a program to design a Push Down Automata for the following language:  $L = \{a^n b^n \text{ where } n > 0\}$

### **Solution:**

---

```
#include <iostream>
#include <stack>
using namespace std;
int main() {
    string input;
    stack<char> stack;
    stack.push('z');
    cout<<"Enter any String:";
    cin>>input;

    for(int i=0;i<=input.size();i++){
        if(input[i]=='a' && (stack.top()=='z' || stack.top()=='A')){
            stack.push('A');
        }
        else if(input[i]=='b' && stack.top()=='A'){
            stack.pop();
        }
        else if(i==input.size() && stack.top()=='z'){
            cout<<"String Accepted"<<endl;
            return 0;
        }
        else{
            cout<<"Invalid String";
            break;
        }
    }
    return 0;
}
```

### **Output:**

Enter any String: aab

Invalid String

Enter any String: aabb

String Accepted

## Program: 11

### **Problem Statement:**

Write a program to design a Push Down Automata for the language:  $L = \{WCWR \mid W = \text{string of 0's and 1's}\}$  C is a special symbol and R is reverse of string.

### **Solution:**

---

```
#include <iostream>

using namespace std;

int main()
{
    char input[100];
    char stack[100];
    int top=-1,i=-1;
    stack[++top]='z';
    cout<<"Enter any String:";
    cin>>input;

    q0:
    i++;
    {
        if(input[i]=='a' && stack[top]=='z'){
            stack[++top]='A';
            //cout<<stack[top];
            goto q0;
        }
        else if(input[i]=='b' && stack[top]=='z'){
            stack[++top]='B';
            goto q0;
        }
        else if(input[i]=='a' && (stack[top]=='A' || stack[top]=='B')){
            stack[++top]='A';
            goto q0;
        }
        else if(input[i]=='b' && (stack[top]=='A' || stack[top]=='B')){
            stack[++top]='B';
            goto q0;
        }
        else if(input[i]=='c' && (stack[top]=='A' || stack[top]=='B')){
            goto q1;
        }
        else{
            goto invalid;
        }
    }
}
```

```

    }

}
q1:
    i++;
    {

        if(input[i]=='a' && stack[top]=='A'){
            --top;
            goto q1;
        }
        else if(input[i]=='b' && stack[top]=='B'){
            --top;
            goto q1;
        }
        else if(input[i]=='\0' && stack[top]=='z'){
            goto valid;
        }
        else{
            goto invalid;
        }
    }
valid:{
    cout<<"\nValid String"<<endl;
    return 0;
}
invalid:{
    cout<<"\nInvalid String"<<endl;
    return 0;
}

return 0;
}

```

**Output:**

Enter any String:ababcbaba

Valid String

Enter any String:abacba

Invalid String