**Q1.** #include<iostream>

#include<string>

using namespace std;

void q0(string w , int i);

void q1(string w , int i);

void q2(string w , int i);

void q3(string w , int i);

int main()

{

string w;

cout << "enter the string (1/0) : ";

cin >> w;

q0(w, 0);

return 0;

}

void q0(string w , int i)

{

if (i == w.length())

cout << "String rejected";

if (w[i] == '0')

q0(w, i + 1);

else if (w[i] == '1')

q1 (w, i + 1);

}

void q1(string w , int i)

{

if (i == w.length())

cout << "String rejected";

if (w[i] == '0')

q0(w, i + 1);

else if (w[i] == '1')

q2 (w, i + 1);

}

void q2(string w , int i)

{

if (i == w.length())

cout << "String rejected";

if (w[i] == '0')

q0(w, i + 1);

else if (w[i] == '1')

q3 (w, i + 1);

}

void q3(string w , int i)

{

if (i == w.length())

cout << "String accepted :)";

else

q3 (w, i + 1);

}

------------------------------------------------------------------------------------------------------------------------------------------

**Q2.** #include<iostream>

#include<string.h>

using namespace std;

void q0(string w , int i);

void q1(string w , int i);

void q2(string w , int i);

void q3(string w , int i);

void q4(string w , int i);

int main()

{

string w;

cout<<"Enter the string(1,0) : ";

cin>>w;

q0(w, 0);

return 0;

}

void q0(string w , int i)

{

if (i == w.length())

cout << "String Rejected";

if (w[i] == '0')

q0(w, i+1);

else if (w[i] == '1')

q1(w, i+1);

}

void q1(string w , int i)

{

if (i == w.length())

cout << "String Rejected";

if (w[i] == '0')

q1(w, i+1);

else if (w[i] == '1')

q2(w, i+1);

}

void q2(string w , int i)

{

if (i == w.length())

cout << "String Accepted";

if (w[i] == '0')

q2(w, i+1);

else if (w[i] == '1')

q3(w, i+1);

}

void q3(string w , int i)

{

if (i == w.length())

cout << "String Accepted";

if (w[i] == '0')

q3(w, i+1);

else if (w[i] == '1')

q4(w, i+1);

}

void q4(string w , int i)

{

if (i == w.length())

cout << "Dead State";

if (w[i] == '0')

q4(w, i+1);

else if (w[i] == '1')

q4(w, i+1);

}

-------------------------------------------------------------------------------------------------------------------------------------------

**Q3.** #include <iostream>

#include <string>

using namespace std;

void q1(string w, int i);

void q2(string w, int i);

void q3(string w, int i);

void q4(string w, int i);

void q5(string w, int i);

void q6(string w, int i);

void q7(string w, int i);

void q8(string w, int i);

void q9(string w, int i);

void q10(string w, int i);

void q11(string w, int i);

void q12(string w, int i);

void q1(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q2(w, i + 1);

}

else if (w[i] == '0')

{

q3(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q2(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q5(w, i + 1);

}

else if (w[i] == '0')

{

q4(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q3(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q7(w, i + 1);

}

else if (w[i] == '0')

{

q6(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q4(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q8(w, i + 1);

}

else if (w[i] == '0')

{

q4(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q5(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q9(w, i + 1);

}

else if (w[i] == '0')

{

q5(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q6(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q6(w, i + 1);

}

else if (w[i] == '0')

{

q10(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q7(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q7(w, i + 1);

}

else if (w[i] == '0')

{

q11(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q8(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q8(w, i + 1);

}

else if (w[i] == '0')

{

q12(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q9(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q12(w, i + 1);

}

else if (w[i] == '0')

{

q5(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q10(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q6(w, i + 1);

}

else if (w[i] == '0')

{

q1(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q11(string w, int i)

{

if (i == w.length())

{

cout << "String Rejected" << endl;

return;

}

if (w[i] == '1')

{

q2(w, i + 1);

}

else if (w[i] == '0')

{

q12(w, i + 1);

}

else

{

cout << "String Rejected" << endl;

}

}

void q12(string w, int i)

{

if (i == w.length())

{

cout << "String Accepted" << endl;

}

else

{

cout << "String Rejected" << endl;

}

}

int main()

{

string w;

cout << "Enter the string (1,0): ";

cin >> w;

q1(w, 0);

return 0;

}

-------------------------------------------------------------------------------------------------------------------------------------------

**Q4.** #include<iostream>

#include<string.h>

using namespace std;

void q0(string w , int i);

void q1(string w , int i);

void q2(string w , int i);

void q3(string w , int i);

int main()

{

string w;

cout << "Enter the string(a,b) : ";

cin >> w;

q0(w, 0);

return 0;

}

void q0(string w , int i)

{

if (i == w.length())

cout << "String Rejected";

if (w[i] == 'a')

q1(w, i + 1);

else if (w[i] == 'b')

q3(w, i + 1);

}

void q1(string w , int i)

{

if (i == w.length())

cout << "String Rejected";

if (w[i] == 'a' || w[i] == 'b')

q2(w, i + 1);

}

void q2(string w , int i)

{

if (i == w.length())

cout << "String Accepted";

if (w[i] == 'a')

q1(w, i + 1);

else if (w[i] == 'b')

q2(w, i + 1);

}

void q3(string w , int i)

{

cout << "Dead State\nString Rejected";

}

-------------------------------------------------------------------------------------------------------------------------------------

**Q5.** #include<iostream>

#include<string.h>

using namespace std;

void q0(string w , int i);

void q1(string w , int i);

void q2(string w , int i);

void q3(string w , int i);

int main()

{

string w;

cout << "Enter the string(a,b) : ";

cin >> w;

q0(w, 0);

return 0;

}

void q0(string w , int i)

{

cout<<"\n"<<w[i]<<" : State 1";

if (i == w.length())

cout << "\nString Accepted";

if (w[i] == 'a')

q2(w, i + 1);

else if (w[i] == 'b')

q1(w, i + 1);

}

void q1(string w , int i)

{

cout<<"\n"<<w[i]<<" : State 2";

if (i == w.length())

cout << "\nString Rejected";

if (w[i] == 'a')

q3(w, i + 1);

else if (w[i] == 'b')

q0(w, i + 1);

}

void q2(string w , int i)

{

cout<<"\n"<<w[i]<<" : State 3";

if (i == w.length())

cout << "\nString Rejected";

if (w[i] == 'a')

q0(w, i + 1);

else if (w[i] == 'b')

q3(w, i + 1);

}

void q3(string w , int i)

{

cout<<"\n"<<w[i]<<" : State 4";

if (i == w.length())

cout << "\nString Rejected";

if (w[i] == 'a')

q1(w, i + 1);

else if (w[i] == 'b')

q2(w, i + 1);

}

-------------------------------------------------------------------------------------------------------------------------------------

**Q6.** #include <iostream>

#include <string>

using namespace std;

void q1(string w, int i);

void q2(string w, int i);

void q3(string w, int i);

void q4(string w, int i);

void q5(string w, int i);

void q1(string w, int i){

if (i==w.length()){

cout << "String Rejected";

return;}

if (w[i]=='a')

q2(w, i+1);

else if (w[i]=='b')

q3(w, i+1);

}

void q2(string w, int i){

if (i==w.length()){

cout<<"String Accepted";

return;}

if (w[i]=='a')

q4(w, i+1);

else if (w[i]=='b')

q3(w, i+1);

}

void q3(string w, int i){

if (i==w.length()){

cout<<"String Accepted";

return;}

if (w[i]=='a')

q2(w, i+1);

else if (w[i]=='b')

q3(w, i+1);

}

void q4(string w, int i){

if (i==w.length()){

cout<<"String Accepted";

return;}

if (w[i]=='a')

q4(w, i+1);

else if (w[i]=='b')

q5(w, i+1);

}

void q5(string w, int i){

if (i==w.length()){

cout<<"String Accepted";

return;}

if (w[i]=='a')

q4(w, i+1);

else if (w[i]=='b')

q5(w, i+1);

}

void r1(string w, int i);

void r2(string w, int i);

void r3(string w, int i);

void r4(string w, int i);

void r5(string w, int i);

void r6(string w, int i);

void r1(string w, int i){

if (i==w.length()){

cout << "String Rejected";

return;}

if (w[i]=='a')

r1(w, i+1);

else if (w[i]=='b')

r2(w, i+1);

}

void r2(string w, int i){

if (i==w.length()){

cout<<"String Accepted";

return;}

if (w[i]=='a')

r3(w, i+1);

else if (w[i]=='b')

r2(w, i+1);

}

void r3(string w, int i){

if (i==w.length()){

cout<<"String Rejected";

return;}

if (w[i]=='a')

r4(w, i+1);

else if (w[i]=='b')

r2(w, i+1);

}

void r4(string w, int i){

if (i==w.length()){

cout<<"String Rejected";

return;}

if (w[i]=='a')

r4(w, i+1);

else if (w[i]=='b')

r5(w, i+1);

}

void r5(string w, int i){

if (i==w.length()){

cout<<"String Accepted";

return;}

if (w[i]=='a')

r6(w, i+1);

else if (w[i]=='b')

r5(w, i+1);

}

void r6(string w, int i){

if (i==w.length()){

cout<<"String Rejected";

return;}

if (w[i]=='a')

r4(w, i+1);

else if (w[i]=='b')

r5(w, i+1);

}

int main(){

string w;

cout<<"\nEnter the string(in a & b):";

cin>>w;

int c;

cout<<"\nL1: Accepting atleast 'aa'\nL2: Ends with b ";

cout<<"\n1. Concatenation of two languages. \n2. Union of two languages";

cin>>c;

if(c==1)

{

q1(w,0);

}

else if(c==2)

{

r1(w,0);

}

return 0;

}

**—-------------------------------------------------------------------------**

**Q7.** #include <iostream>

#include <string>

#include <stack>

using namespace std;

class PDA {

private:

stack<char> st;

void pushToStack() {

st.push('a');

cout << "Pushed 'a' to the stack\n";

}

void popFromStack() {

if (!st.empty()) {

st.pop();

cout << "Popped 'a' from the stack\n";

}

}

void checkString(string &s, int index) {

if (index == s.length()) {

if (st.empty()) {

cout << "\nString Accepted\n";

} else {

cout << "\nString Rejected (Stack not empty)\n";

}

return;

}

char ch = s[index];

if (ch == 'a') {

pushToStack();

checkString(s, index + 1);

}

else if (ch == 'b') {

if (st.empty()) {

cout << "\nString Rejected (No 'a' for 'b')\n";

return;

} else {

popFromStack();

checkString(s, index + 1);

}

} else {

cout << "\nString Rejected (Invalid character)\n";

return;

}

}

public:

void process(string s) {

while (!st.empty()) {

st.pop();

}

cout << "Checking string: " << s << endl;

checkString(s, 0);

}

};

int main() {

string s;

PDA pda;

cout << "Enter a string of 'a' and 'b': ";

cin >> s;

pda.process(s);

return 0;

}

**—-------------------------------------------------------------------------**

**Q8.** #include <iostream>

#include <string>

#include <stack>

using namespace std;

class PDA {

private:

stack<char> st;

void pushToStack(char c) {

st.push(c);

cout << "Pushed '" << c << "' to the stack\n";

}

void popFromStack() {

if (!st.empty()) {

char popped = st.top();

st.pop();

cout << "Popped '" << popped << "' from the stack\n";

}

}

void processString(string &s, int index, bool isFirstPhase) {

if (index == s.length()) {

if (!isFirstPhase && st.empty()) {

cout << "\nString Accepted\n";

} else if (isFirstPhase) {

cout << "First phase completed. Now, waiting for X.\n";

} else {

cout << "\nString Rejected (Mismatch or Stack is not empty)\n";

}

return;

}

char currentChar = s[index];

if (isFirstPhase) {

if (currentChar == 'X') {

processString(s, index + 1, false);

} else {

pushToStack(currentChar);

processString(s, index + 1, true);

}

} else {

if (st.empty()) {

cout << "\nString Rejected (More characters in w^r than w)\n";

return;

}

char topChar = st.top();

if (topChar == currentChar) {

popFromStack();

processString(s, index + 1, false);

} else {

cout << "\nString Rejected (Mismatch between w and w^r)\n";

return;

}

}

}

public:

void process(string s) {

while (!st.empty()) {

st.pop();

}

cout << "Checking string: " << s << endl;

processString(s, 0, true);

}

};

int main() {

string s;

PDA pda;

cout << "Enter a string of the form wXw^r (e.g., abXba): ";

cin >> s;

pda.process(s);

return 0;

}

**Q9.**

**#include <iostream>**

**#include <string>**

**class TuringMachine {**

**public:**

**TuringMachine(const std::string& input) : tape(input), head(0) {}**

**bool accept() {**

**// Start processing the tape**

**return process();**

**}**

**private:**

**std::string tape;**

**int head;**

**bool process() {**

**// Step 1: Replace 'a' with 'X'**

**while (head < tape.size() && tape[head] == 'a') {**

**tape[head] = 'X'; // Mark 'a' as 'X'**

**head++;**

**if (!findAndReplace('b', 'Y')) return false; // Find and replace 'b'**

**if (!findAndReplace('c', 'Z')) return false; // Find and replace 'c'**

**head = 0; // Reset head to start for the next iteration**

**}**

**// Step 2: Check if all characters are replaced**

**for (char ch : tape) {**

**if (ch != 'X' && ch != 'Y' && ch != 'Z') {**

**return false; // If any character is left, reject**

**}**

**}**

**return true; // Accepted**

**}**

**bool findAndReplace(char target, char replacement) {**

**while (head < tape.size()) {**

**if (tape[head] == target) {**

**tape[head] = replacement; // Replace target with replacement**

**return true; // Found and replaced**

**}**

**head++;**

**}**

**return false; // Target not found**

**}**

**};**

**int main() {**

**std::string input;**

**std::cout << "Enter a string of the form a^n b^n c^n: ";**

**std::cin >> input;**

**TuringMachine tm(input);**

**if (tm.accept()) {**

**std::cout << "Accepted\n";**

**} else {**

**std::cout << "Rejected\n";**

**}**

**return 0;**

**}**

**Q10.**

**#include <iostream>**

**#include <string>**

**class TuringMachine {**

**public:**

**TuringMachine(const std::string& input) : tape(input), head(input.size() - 1) {}**

**void increment() {**

**// Start processing the tape from the rightmost bit**

**while (head >= 0) {**

**if (tape[head] == '0') {**

**tape[head] = '1'; // Change '0' to '1'**

**return; // Increment done, exit**

**} else if (tape[head] == '1') {**

**tape[head] = '0'; // Change '1' to '0' (carry)**

**head--; // Move left**

**} else {**

**std::cerr << "Invalid character on tape. Only '0' and '1' are allowed." << std::endl;**

**return;**

**}**

**}**

**// If we exit the loop, it means we have a carry out**

**tape = '1' + tape; // Add '1' at the beginning**

**}**

**std::string getTape() const {**

**return tape;**

**}**

**private:**

**std::string tape;**

**int head;**

**};**

**int main() {**

**std::string input;**

**std::cout << "Enter a binary number: ";**

**std::cin >> input;**

**// Validate input**

**for (char ch : input) {**

**if (ch != '0' && ch != '1') {**

**std::cerr << "Invalid input. Please enter a binary number." << std::endl;**

**return 1;**

**}**

**}**

**TuringMachine tm(input);**

**tm.increment();**

**std::cout << "Incremented binary number: " << tm.getTape() << std::endl;**

**return 0;**

**}**