Exp 24:-

Design TM using Simulator to accept the input String WW

Aim:-

To design TM using Simulator to accept the input String WW

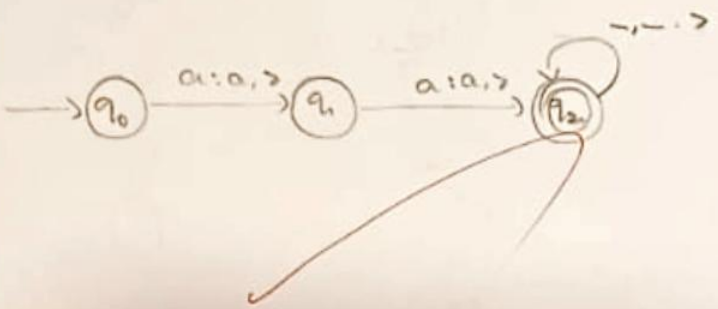
Apparatus:-

Autosim

Input:-

To Check the String WW

Output :-



Result:-

We designed TM to accept the input String WW by using autosim.

Exp 11 :-

Design TM using Simulator to accept the input String $a^n b^n$.

Aim:-

To design TM to accept the input String $a^n b^n$

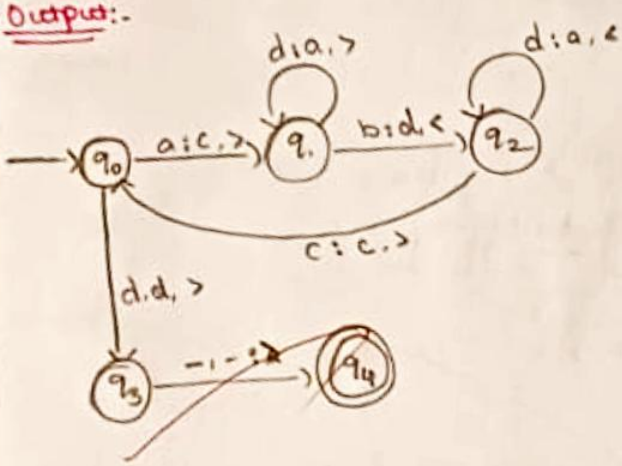
Apparatus:-

Actto Sim.

Input:-

To check the String $a^n b^n$

Output:-



Result:-

We designed TM to accept the input String $a^n b^n$ by using autosim.

Exp 23:-

Design TM using Simulator to accept the input string palindrome ababa.

Aim:-

To design TM using Simulator to accept the input string ababa

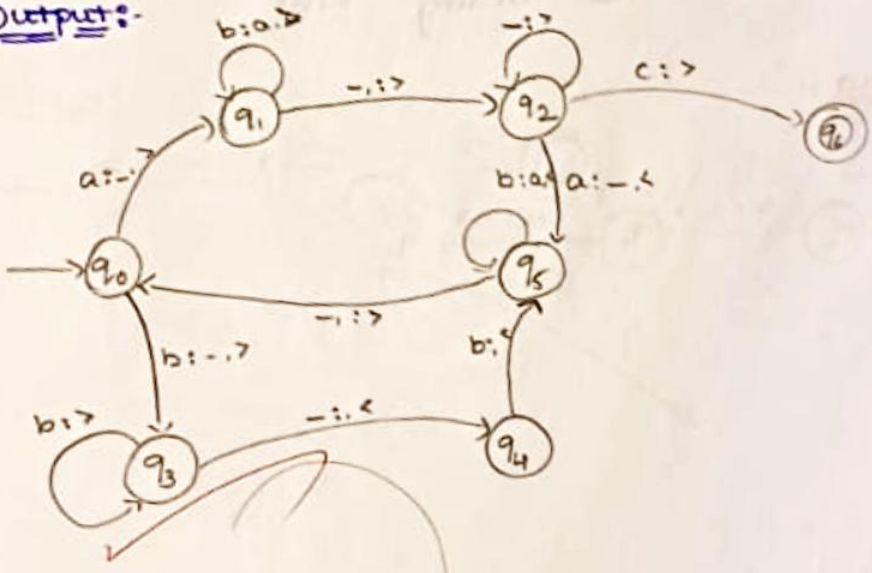
Apparatus:-

Autosim

Input:-

To check the string ababa.

Output:-



Result:-

We designed TM to accept the input string ababa by using autosim.

Exp 4:-

Design DFA using Simulator to accept the string end with ab over set {a, b}

W = aaabab.

Aim:-

To design DFA that the string end with ab to accept the string aaabab

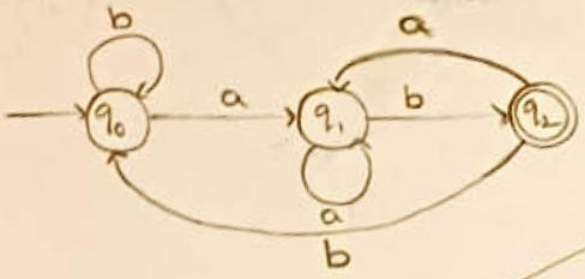
Apparatus:-

Auto sim

Input :-

The string is aaabab

Output :-



Result:

We designed the DFA that accept the string "aaabab" using auto sim

Exp 2:-

Design DFA using Simulator to accept Even number of a's

Aim:-

To design a DFA to accept the Even number of a's

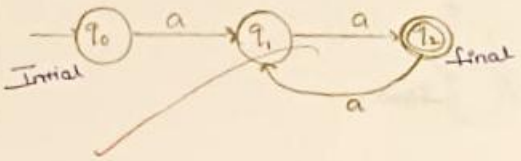
Apparatus:-

Auto Sim

Input :-

Even number of a's like aa, aaaa, aaaaaa ....

output :-



Result:-

We designed the DFA that accept the input Even number of a's using auto Sim.

Exp 3:-

Design DFA using Simulator to accept odd number of a's

Aim:-

To design the DFA using Simulator to accept odd number of a's

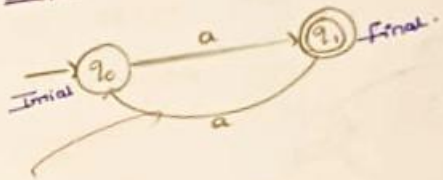
Apparatus:-

Auto Sim

Input:-

odd number of a's like a, aaa, aaaaa .....

output:-



Result:-

We designed the DFA that accept the input odd number of a's using auto Sim.

Exp16:-

Design NFA using Simulator to accept the input String "bbc", "c" and "bcaaa".

Aim:-

To design NFA to accept the input String "bbc", "c" and "bcaaa"

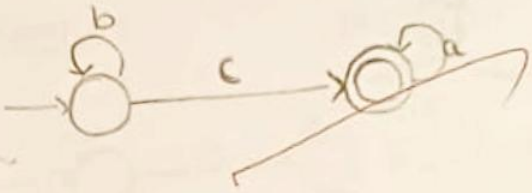
Apparatus:-

Automata sim

Input:-

To check the Strings "bbc", "c" and "bcaaa".

Output:-



Result:-

We designed the NFA to accept the input String "bbc", "c" and "bcaaa".

Exp 18:-

Design PDA using Simulator to accept the input string $a^n b^n$.

Aim:-

To design PDA using Simulator to accept the input string $a^n b^n$.

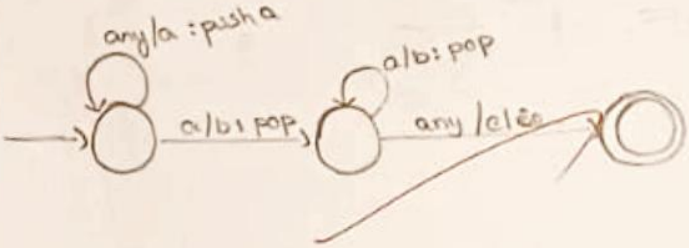
Apparatus:-

Auto Sim

Input:-

To check the input string $a^n b^n$.

Output:-



Result:-

Thus designed PDA to accept the input string $a^n b^n$.

Design NFA using Simulator to accept the String that Starts and end with different Symbols over the input {a, b}

Aim:

To design NFA to accept the String that Starts and end with different Symbols

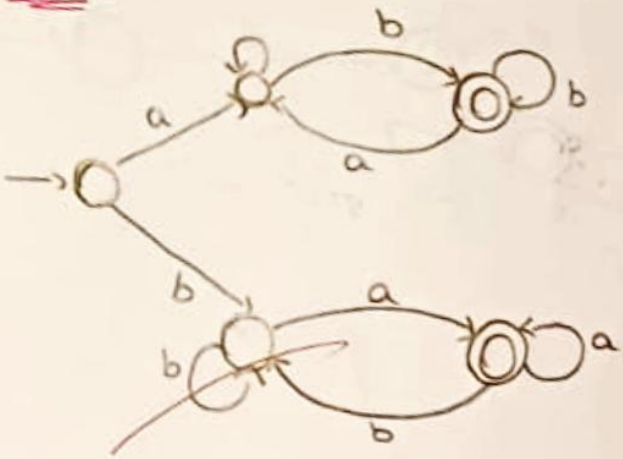
Apparatus:

Atoro sim

Input:

The String that Starts and end with different Symbols.

Output:



Result:

We designed NFA to accept the Strings that Starts and end with different symbols

Exp 3:-

Design DFA using Simulator to accept String "a", "ac", and "bac".

Aim:

To design the DFA to accept the input String "a", "ac" and "bac".

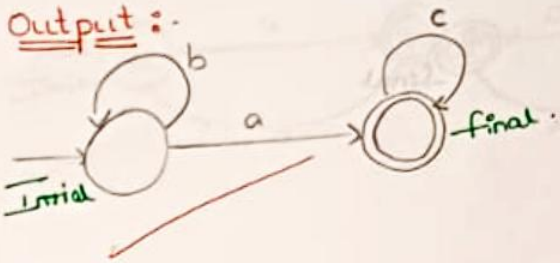
Apparatus :-

Auto Simulator

Input :-

"a", "ac" and "bac"

Output :-



Result :-

We designed the DFA that accept the String "a", "ac" and "bac" using autosim.

Exp 6:
Design DFA using Simulator to accept the String Start with a (or) b over the set {a, b}.

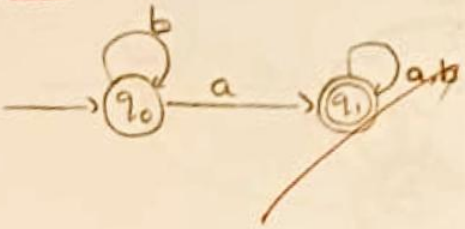
Aim:

To design DFA to accept the String Start with a (or) b

Apparatus:
Auto sim

Input:
The String Starts with "a" or "b"
a babaaa.

Output:



Result:
We designed the DFA to accept the String Start with a (or) b using Auto sim.

Design DFA using Simulator to accept the String having "abc" as Substring over the set {a, b, c}

Aim:-

To design DFA to accept the String having "abc" as Substring over the set {a, b, c}

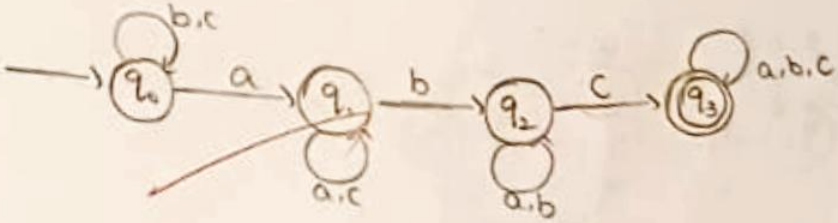
Apparatus:-

Automata sim

Input:-

The input String having "abc" as Substring

Output:-



Result:-

We designed DFA to accept the String having "abc" as Substring using Automata sim.

Design DFA using Simulator to accept Strings in which a's always appear tripled over input {a, b}

Aim:-

To design DFA to accept Strings in which a's always appear tripled.

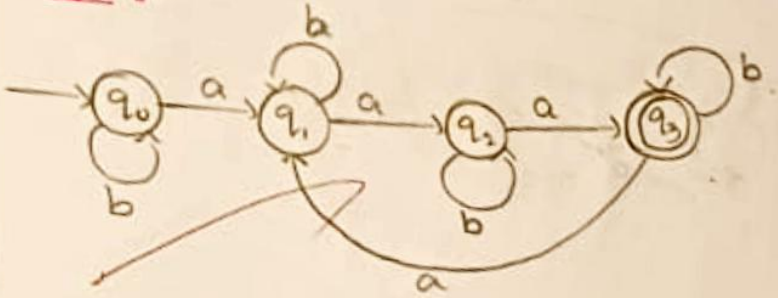
Apparatus:-

Auto sim.

Input:-

The String in which a's always appear tripled.

Output:-



Result:-

We designed DFA to accept Strings in which a's always appear tripled.

Exp 13:-

Design NFA to accept any number of a's where input = {a, b}

Aim:-

To design NFA to accept any number of a's

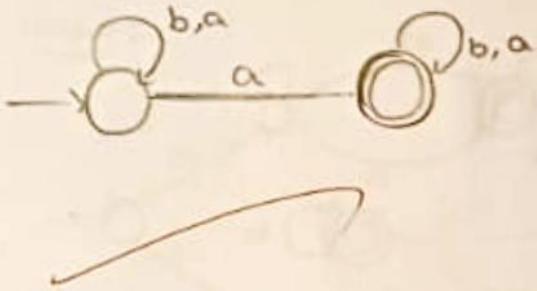
Apparatus:-

Auto Sim

Input:-

The String that accept any number of a's

Output:-



Result:-

We designed NFA to accept the String any number of a's by using autosim.