

Exp1: Write a program to implement a Deterministic Automation

Answer:

Source Code:

```
#include <iostream>
#include <sstream>
#include <string>
#include <fstream>

using namespace std;

string valid_check(int dfa[][100], int final_states[], int final_states_count, int initial_state)
{
    string str;
    cout << "Enter string: ";
    cin >> str;

    int curr_state = dfa[initial_state][str[0] - '0'];
    int i = 1;
    while (i < str.size() && curr_state != -1)
    {
        curr_state = dfa[curr_state][str[i++] - '0'];
    }

    for (i = 0; i < final_states_count; i++)
    {
        if (final_states[i] == curr_state)
            return "ACCEPTED";
    }
}
```

```

    return "NOT ACCEPTED";
}

int main()
{
    ifstream file;

    int final_states[100], dfa[100][100], final_states_count = 0, initial_state;

    file.open("./DFA.txt");

    int n = 0, m = 0;

    bool counting = true;

    string str, temp;
    getline(file, str);
    initial_state = stoi(str);
    getline(file, str);
    stringstream X(str);
    while (getline(X, temp, ' '))
    {
        final_states[final_states_count++] = stoi(temp);
    }

    while (getline(file, str))
    {
        stringstream X(str);
        m = 0;
        while (getline(X, temp, ' '))
        {
            dfa[n][m++] = stoi(temp);

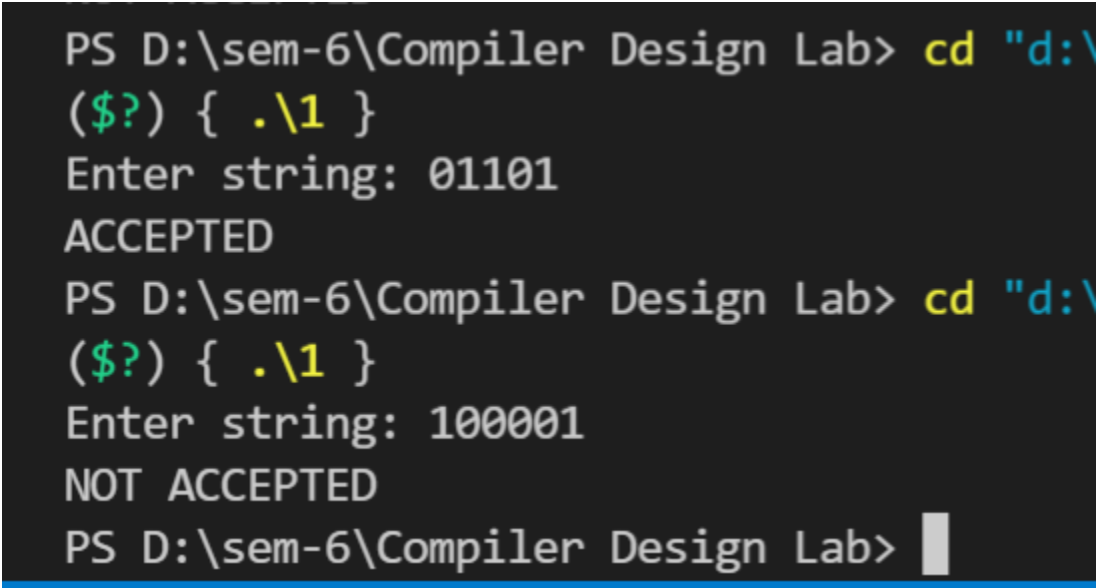
```

```
    }  
    n++;  
}  
  
cout << valid_check(dfa, final_states, final_states_count, initial_state);  
  
return 0;  
}
```

File Input:

```
0  
2 4  
1 2  
-1 4  
3 -1  
1 4  
3 2
```

Output:



```
PS D:\sem-6\Compiler Design Lab> cd "d:\sem-6\Compiler Design Lab"  
($?) { .\1 }  
Enter string: 01101  
ACCEPTED  
PS D:\sem-6\Compiler Design Lab> cd "d:\sem-6\Compiler Design Lab"  
($?) { .\1 }  
Enter string: 100001  
NOT ACCEPTED  
PS D:\sem-6\Compiler Design Lab> 
```