Name: Bhavya Patel

Roll No: 20BCE198

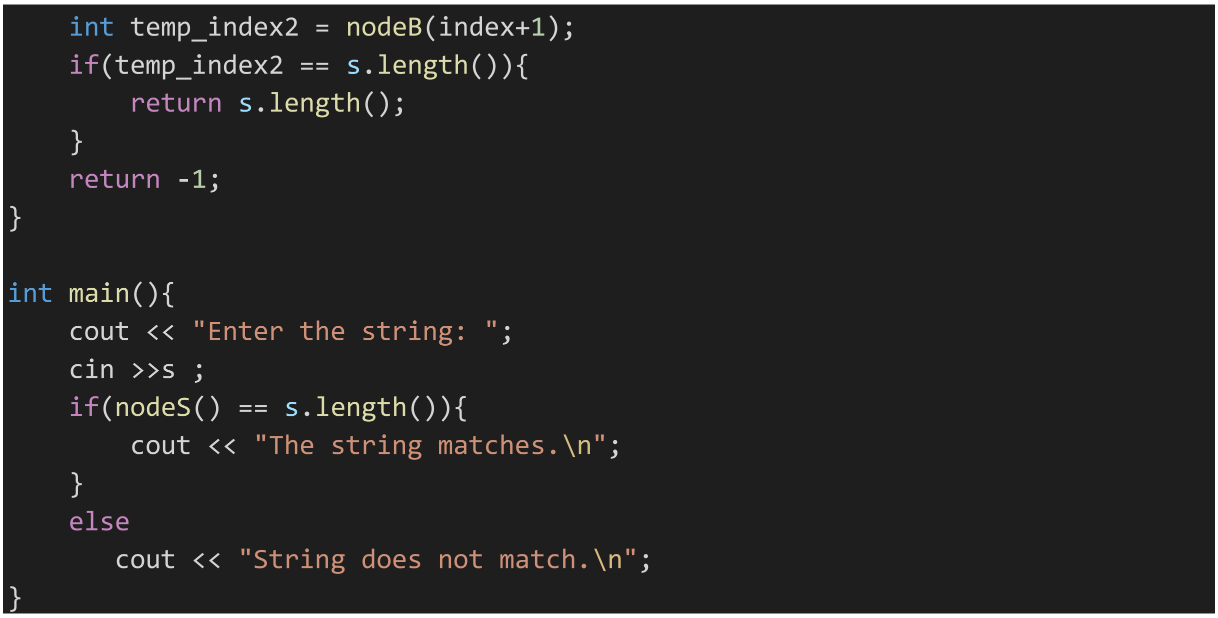
Course: Compiler construction

Practical No: 2

Aim: To implement a Recursive Descent Parser Algorithm for the grammar.

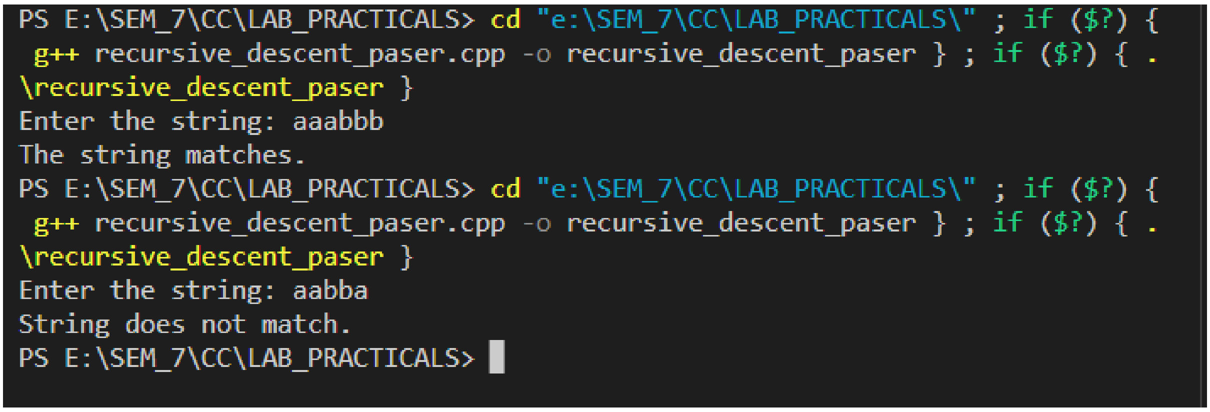
Methodology followed:

```
#include<bits/stdc++.h>

using namespace std;

string s;

int nodeA(int index){
    if(index == s.length()){
        return index;
    }
    if(s[index]=='a'){
        return nodeA(index+1);
    }
    return index;
}

int nodeB(int index){
    if(index == s.length()){
        return index;
    }
    if(s[index] == 'b'){
        if(nodeB(index+1) == s.length()){
            return s.length();
        }
    }
    return index;
}

int nodeS(int index = 0){
    if(s[index] != 'a')
        return -1;

    int temp_index1 = nodeA(index+1);

    if(nodeB(temp_index1) == s.length()){
        return s.length();
    }
```



```
    int temp_index2 = nodeB(index+1);
    if(temp_index2 == s.length()){
        return s.length();
    }
    return -1;
}

int main(){
    cout << "Enter the string: ";
    cin >>s ;
    if(nodeS() == s.length()){
        cout << "The string matches.\n";
    }
    else
       cout << "String does not match.\n";
}
```

Output:



```
PS E:\SEM_7\CC\LAB_PRACTICALS> cd "e:\SEM_7\CC\LAB_PRACTICALS\" ; if ($?) {
 g++ recursive_descent_paser.cpp -o recursive_descent_paser } ; if ($?) { .
\recursive_descent_paser }
Enter the string: aaabbb
The string matches.
PS E:\SEM_7\CC\LAB_PRACTICALS> cd "e:\SEM_7\CC\LAB_PRACTICALS\" ; if ($?) {
 g++ recursive_descent_paser.cpp -o recursive_descent_paser } ; if ($?) { .
\recursive_descent_paser }
Enter the string: aabba
String does not match.
PS E:\SEM_7\CC\LAB_PRACTICALS> █
```