Roll No: 20BCE204
Course: 2CSDE93 - Blockchain Technology
Practical No: 4
Aim: Practical byzantine fault tolerance in Blockchain systems, the math behind this algorithm.

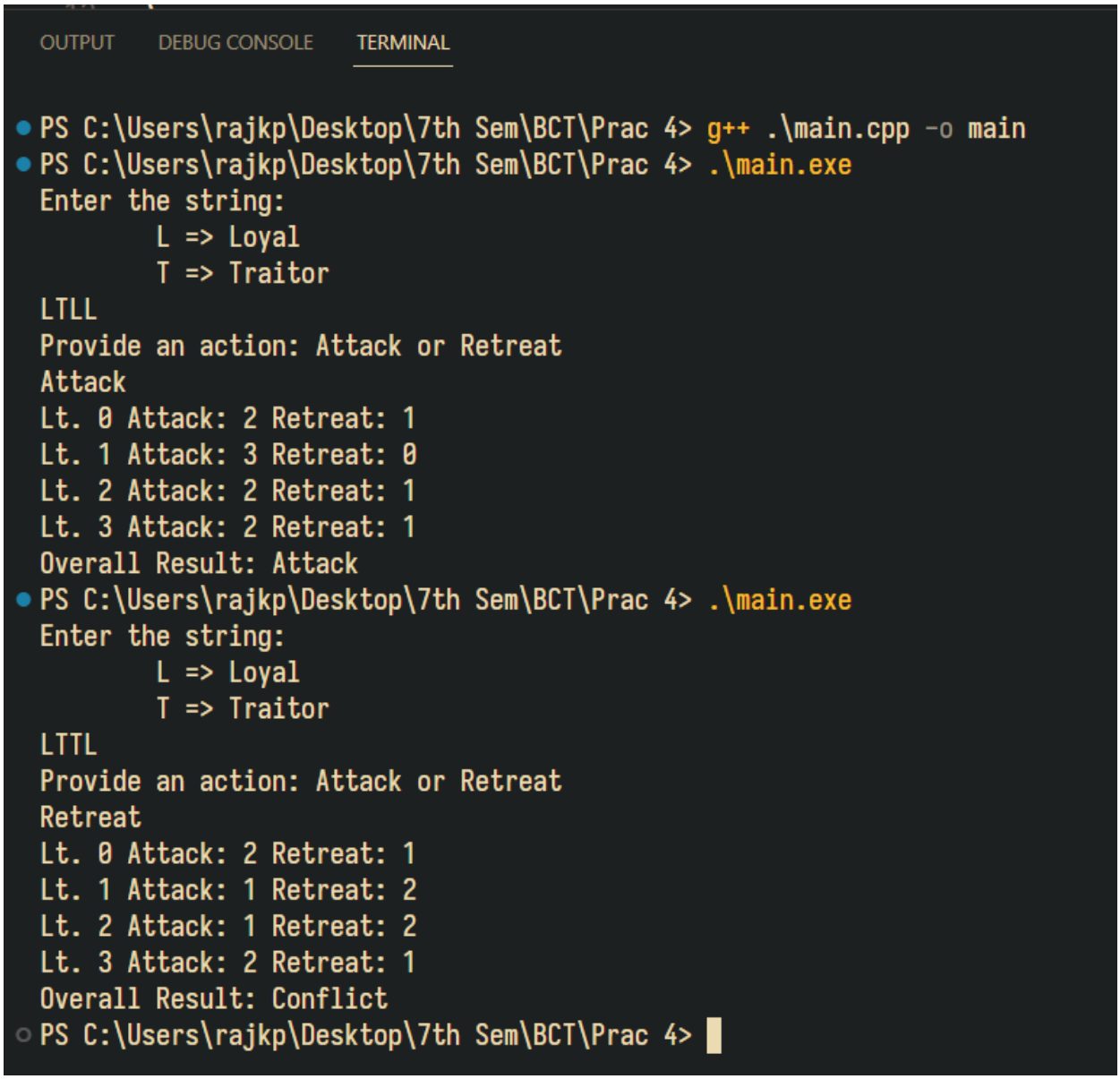
Code:

```
#include <bits/stdc++.h>
using namespace std;

enum Result
{
    Attack,
    Retreat,
    Conflict
};

int main()
{
    cout << "Enter the string: \n";
    cout << "\tL => Loyal\n";
    cout << "\tT => Traitor\n";
    string input = "LLLL";
    for (int i = 0; i < 4; i++)
    {
        cin >> input[i];
    }

    cout << "Provide an action: Attack or Retreat\n";
    string s_action;
    cin >> s_action;
    Result action = Result::Conflict;

    if (s_action == "Attack")
```

```
        action = Result::Attack;
    else if (s_action == "Retreat")
        action = Result::Retreat;
    else
    {
        cout << "ERROR: Incorrect input\n";
        exit(1);
    }
    map<int, vector<Result>> res;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            if (i == j)
                continue;
            if (input[i] == 'T')
            {
                if (action == Result::Attack)
                    res[j].push_back(Result::Retreat);
                else
                    res[j].push_back(Result::Attack);
            }
            if (input[i] == 'L')
            {
                res[j].push_back(action);
            }
        }
    }

    int overall_attack = 0, overall_retreat = 0;
    for (int i = 0; i < 4; i++)
    {
        Result ans = Result::Conflict;
        int attack = 0;
        int retreat = 0;
        for (auto it : res[i])
```

```
        {
            if (it == Result::Attack)
            {
                attack++;
                overall_attack++;
            }
            if (it == Result::Retreat)
            {
                retreat++;
                overall_retreat++;
            }
        }
        cout << "Lt. " << i << " ";
        cout << "Attack: " << attack << " Retreat: " << retreat
<< endl;
    }

    if (overall_attack > overall_retreat)
    {
        cout << "Overall Result: Attack\n";
    }
    else if (overall_attack < overall_retreat)
    {
        cout << "Overall Result: Retreat\n";
    }
    else
    {
        cout << "Overall Result: Conflict\n";
    }

    return 0;
}
```

Output:


OUTPUT
DEBUG CONSOLE
TERMINAL
PS C:\Users\rajkp\Desktop\7th Sem\BCT\Prac 4> g++ .\main.cpp -o main
PS C:\Users\rajkp\Desktop\7th Sem\BCT\Prac 4> .\main.exe
Enter the string:
L => Loyal
T => Traitor
LTLL
Provide an action: Attack or Retreat
Attack
Lt. 0 Attack: 2 Retreat: 1
Lt. 1 Attack: 3 Retreat: 0
Lt. 2 Attack: 2 Retreat: 1
Lt. 3 Attack: 2 Retreat: 1
Overall Result: Attack
PS C:\Users\rajkp\Desktop\7th Sem\BCT\Prac 4> .\main.exe
Enter the string:
L => Loyal
T => Traitor
LTTL
Provide an action: Attack or Retreat
Retreat
Lt. 0 Attack: 2 Retreat: 1
Lt. 1 Attack: 1 Retreat: 2
Lt. 2 Attack: 1 Retreat: 2
Lt. 3 Attack: 2 Retreat: 1
Overall Result: Conflict
PS C:\Users\rajkp\Desktop\7th Sem\BCT\Prac 4>