Experiment 1.2

Student Name: Rohan Jaiswal
UID: 21BCS2856
Branch: BE-CSE
Section/Group: KRG-CC-1-B
Semester: 6th
Date of Performance: 23-01-2024
Subject Name: Advance Programming-2
Subject Code: 21CSP-251

1. Aim:

- To Solve the Rotate String.
- To Solve the Repeated String Match.

2. Objective:

- Given two strings s and goal, return true if and only if s can become goal after some number of shifts on s.
  A shift on s consists of moving the leftmost character of s to the rightmost position.
- Given two strings a and b, return the minimum number of times you should repeat string a so that string b is a substring of it. If it is impossible for b to be a substring of a after repeating it, return -1.

3. Algo. /Approach and output:

1st:

```
class Solution {
bool solve(queue<int> queOne, queue<int> queTwo, int size){
   while(size--)
   {
      int front = queTwo.front();
      queTwo.pop();
      queTwo.push(front);
      if(queOne == queTwo){
         return true;
      }
   }
   return false;
```

```
}
public:
    bool rotateString(string s, string goal) {
        queue<int> queOne;
        queue<int> queTwo;
        if(s.size() != goal.size()){
            return false;
        }
        for(int i = 0; i < s.size(); i++){
            queOne.push(s[i]);
        }
        for(int i = 0; i < goal.size(); i++){
            queTwo.push(goal[i]);
        }
        int size = goal.size();
        return solve(queOne, queTwo, size);
    }
};
```

2nd:

```
class Solution {
public:
```

```
int solve(string &a, string &b){
    string s = a;
    int res = 1;
    int n = b.size()/a.size();
    for(int i=0;i<=n+1;i++){
        if(s.find(b) != string::npos) return res;
            s+=a;
            res++;
    }
    return -1;
}

    int repeatedStringMatch(string a, string b) {
        return solve(a, b);
    }
};
```