3. 1232. Check If It Is a Straight Line

You are given an array coordinates, coordinates[i] = [x, y], where [x, y] represents the coordinate of a point. Check if these points make a straight line in the XY plane.

```
//Time Complexity:O(n)

class Solution {

public:

    bool checkStraightLine(vector<vector<int>>& coordinates) {

        int points=coordinates.size();

        int xdiff=coordinates[1][0]-coordinates[0][0];

        int ydiff=coordinates[1][1]-coordinates[0][1];

        int curr_Xdiff, curr_Ydiff;

        for(int i=2;i<points;++i)

        {

            curr_Xdiff=coordinates[i][0]-coordinates[i-1][0];

            curr_Ydiff=coordinates[i][1]-coordinates[i-1][1];


            if(xdiff*curr_Ydiff!=ydiff*curr_Xdiff)

            {

                return false;

            }

        }

        return true;

    }

};
```

4. 844. Backspace String Compare

Given two strings s and t, return true if they are equal when both are typed into empty text editors. '#' means a backspace character.

Note that after backspacing an empty text, the text will continue empty.

```
//Time Complexity: O(n)

//Space Complexity:O(n)

class Solution {

public:

    bool backspaceCompare(string s, string t) {

    stack<int> s1,s2;

    string str1, str2;

    for(int i=0;i<s.size();i++)

    {

        if(s[i]=='#' && !s1.empty())

        {

            s1.pop();

        }

        else if(s[i]!='#')



        {

            s1.push(s[i]);

        }

    }

     for(int i=0;i<t.size();i++)
    {
        if(t[i]=='#' && !s2.empty())
        {
            s2.pop();
        }
        else if(t[i]!='#')
        {
            s2.push(t[i]);
        }


    }
    while(!s1.empty())
    {
        str1.push_back(s1.top());
        s1.pop();
    }
      while(!s2.empty())
    {
        str2.push_back(s2.top());
        s2.pop();
    }
     return str1==str2;
    }
};
```

5. 796. Rotate String

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.

- For example, if s = "abcde", then it will be "bcdea" after one shift.

```
class Solution {

    public boolean rotateString(String s, String goal) {

        return s.length()==goal.length() && (s+s).contains(goal);

    }

}
```

6. 415. Add Strings

Given two non-negative integers, num1 and num2 represented as string, return the sum of num1 and num2 as a string.

```
//Time Complexity: O(n)

class Solution {

public:

    string addStrings(string num1, string num2) {

        int a=num1.size()-1;

        int b=num2.size()-1;

        int carry=0;

        string ans;



        while(a>=0 || b>=0 ||carry==1)

        {

            if(a>=0)

            {

                carry+=num1[a]-'0';

                a--;



            }

            if(b>=0)

            {

                carry+=num2[b]-'0';

                b--;
            }
            ans+=carry%10+'0';
            carry=carry/10;
        }
        reverse(ans.begin(),ans.end());
        return ans;

    }
};
```