Flip Game I

You are playing the following Flip Game with your friend: Given a string that contains only these two characters:+ and -, you and your friend take turns to flip two **consecutive** "++" into "--". The game ends when a person can no longer make a move and therefore the other person will be the winner.
Write a function to compute all possible states of the string after one valid move.
For example, given s = "++++", after one move, it may become one of the following states:

```
[
  "--++",
  "+--+",
  "++--"
]
```

If there is no valid move, return an empty list []

Flip Game II

You are playing the following Flip Game with your friend: Given a string that contains only these two characters: + and -, you and your friend take turns to flip two**consecutive** "++" into "--". The game ends when a person can no longer make a move and therefore the other person will be the winner.
Write a function to determine if the starting player can guarantee a win.
For example, given s = "++++", return true. The starting player can guarantee a win by flipping the middle "++" to become "+--+".

Follow up:

Derive your algorithm's runtime complexity.

```
#include<sstream>
#include<iostream>
#include<algorithm>
#include<string>
#include<vector>
#include<map>
#include<stdio.h>
using namespace std;

void flip_dfs(string s, int begin, vector<string> &result)
{
	if(begin>=s.size()) return;
	for(int i=begin;i<s.size()-1;i++)
	{
```

```
            if(s[i]=='+' && s[i]==s[i+1])
            {
                string temp = s.substr(0,i)+"--"+s.substr(i+2);
                result.push_back(temp);
            }else{
                flip_dfs(s,i+1,result);
            }
        }
}

vector<string> flipGame1(string s)
{
        vector<string> res;
        if(s.size()<2) return res;
        flip_dfs(s,0,res);
        return res;
}

bool canWin(string s)
{
        for(int i=0;i<s.size()-1;i++)
        {
            if(s[i]==s[i+1] && s[i]=='+')
            {
                s[i]='-';
                s[i+1]='-';
                bool win = !canWin(s);
                s[i]='+';
                s[i+1]='+';
                if(win) return true;
            }
        }
        return false;
}

bool flipGame2(string s)
{
```

```
        return canWin(s);
}
int main(int argc,char *argv[])
{
        string testcase[] = {"+","-","++","--","+++","++++","+--+","-+-+","+-+-"};
        int sz = (int)sizeof(testcase)/sizeof(testcase[0]);
        vector<string> res;
        //flip game 1
        for(int k=0;k<sz;++k)
        {
                res = flipGame1(testcase[k]);
                for(int j=0;j<res.size();++j)
                        cout<<res[j]<<endl;
                cout<<"¥n";
                res.clear();
        }
        //flip game 2
        for(int k=0;k<sz;++k)
        {
                bool ans = flipGame2(testcase[k]);
                cout << ans << endl;
        }
        return 0;
}
```