动态规划 区间

```
#include <iostream>
#include <vector>
#include <string>
#include <cstring>
#include <set>
#include <map>
#include <queue>
#include <ctime>
#include <random>
#include <sstream>
#include <numeric>
#include <stack>
#include <stdio.h>
#include <algorithm>
using namespace std;

#define Multiple_groups_of_examples
#define rep(i,x,n) for(int i = x; i <= n; i++)
#define vf first
#define vs second

typedef long long LL;
typedef pair<int,int> PII;

const int INF = 0x3f3f3f3f;
const int N = 1e2 + 21;
namespace golitter {
namespace interval {

string str;
int f[N][N];
void inpfile();

void solve() {
    // 求括号串的最少添加数
    // https://blog.csdn.net/weixin_43517157/article/details/106093699
char ph[N];
    while(cin>>ph + 1) {
        if(ph[1] == '0') break;
        int n = strlen(ph + 1);
        vector<vector<int>> f(n+1, vector<int>(n+1));
        for(int len = 2; len <= n; ++len) {
            for(int i = 1;  i + len - 1 <= n; ++i) {
                int j = i + len - 1;
                if(ph[i] == '(' && ph[j] == ')' || ph[i] == '[' && ph[j] == ']')
f[i][j] = f[i+1][j-1] + 2;
                for(int k = i; k < j; ++k) {
                    f[i][j] = max(f[i][j], f[i][k] + f[k+1][j]);
                }
            }
        }
        cout<<f[1][n]<<endl;
    }
}
}}
```