**★★☆☆☆**

**題組：**Problem Set Archive with Online Judge

**題號：**10313: Pay the Price

**解題者：王劭陽**

**解題日期：**2014**年**6**月**5**日**

**題意：**

給定一個非負整數n，要求把n拆成m個正整數之和，求有多少種方法。（輸入資料可能要求m在一個給定的範圍內，否則m任意）

**題意範例：**

4 -> 5 (1+1+1+1),(1+1+2),(2+2),(1+3),(4)

4 1 2 -> 4

**解法：**

動態規劃。設f(i, j)表示把整數i拆分，並且最大加數不超過j的方案數。

 如f(4, 2) = 3，分別是(1+1+1+1),(1+1+2),(2+2)。

 方程如下

 f(i, j) = f(i - j, j) + f(i, j - 1)

 邊界條件f(0, j) = 1

Ferrers圖：一個從上而下的n層格子，上層的格子數不少于下層的格子數時，稱之為Ferrers圖。它有如下性質：第一行与第一列互換，第二行与第二列互換，……，即下圖繞虛線軸翻轉所得的圖仍然是Ferrers圖。

利用Ferrers圖可以得到非常有趣的結果：整數n拆分成k個數的和的方案數，和n拆分成最大數為k的方案數相等。因為n拆分成k個數的和的方案可用k行的圖像表示，翻轉后的圖最上面一行有k個格子。如：

左圖 24=6+6+5+4+3 5個數，最大數為6

右圖 24=5+5+5+4+3+2 6個數，最大數為5

所以根據以上結論，f(i, j)就等於把整數i拆分成不超過j個整數的方案數。

**解法範例：**

無

**討論：**

無

**程式：**

#include <stdio.h>

#include <memory.h>

#define N 300

int main() {

 long long f[N + 1][N + 1];

 memset(f, 0, sizeof(f));

  for (int j = 0; j <= N; j++) {

 f[0][j] = 1;

 }

 for (int i = 1; i <= N; i++) {

 for (int j = 1; j <= N; j++) {

 f[i][j] = f[i][j - 1];

 if (i - j >= 0) {

 f[i][j] += f[i - j][j];

 }

 }

 }

 char s[50];

 while (gets(s)) {

 int n, l1, l2;

 int cnt = sscanf(s, "%d%d%d", &n, &l1, &l2);

 if (l1 > 300) {

 l1 = 300;

 }

 if (l2 > 300) {

 l2 = 300;

 }

 long long ans = 0;

 switch (cnt) {

 case 1 :

 ans = f[n][n];

 break;

 case 2 :

 ans = f[n][l1];

 break;

 case 3 :

 if (n == 0 && l1 == 0) {

 ans = 1;

 break;

 }

 if (l1 == 0) {

 l1 = 1;

 }

 ans = f[n][l2] - f[n][l1 - 1];

 break;

 }

 printf("%lld\n", ans);

 }

}