DP #3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题目名称 | 绵羊 | 分组 | 立方 | 最长公共子序列 |
| 输入文件名 | sheep.in | group.in | cube.in | lcs.in |
| 输出文件名 | sheep.out | group.out | cube.out | lcs.out |
| 每个测试点时限 | 1s | 1s | 1s | 1s |
| 测试点数目 | 10 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 | 10 |
| 内存限制 | 256M | 256M | 256M | 64M |
| 是否有部分分 | 否 | 否 | 否 | 否 |
| 题目类型 | 传统 | 传统 | 传统 | 传统 |

注意：代码长度限制均为64K，不开O2。

1 **绵羊**(sheep.c/cpp/pas)

1.1 **题目描述**

T组数据，对于每一组数据，有n+1个格子从0到n标号，绵羊从0号结点开始，每次若在 x 位置掷骰子，令掷出的数为num，则跳到 x+num 处。另外还有 m 个弹簧，绵羊跳到一个有弹簧的格子上时，不需要掷骰子便可向右跳到某个位置（若此时仍有弹簧将继续向右跳），直到到达 n 或者超出 n 停止。

询问绵羊掷骰子的期望次数。

1.2 **输入格式**

第一行为一个整数T，表示数据组数。

接下来对于每组数据：首先一行两个数n, m，含义如题目所述，接下来每行两个数 a, b，表示在格子 a 处有一个弹簧，将跳到格子 b 上(保证b>a)。

1.3 **输出格式**

输出T行，每行一个实数，表示保留两位小数后的答案。

1.4 **样例输入**

2

2 0

8 3

2 4

4 5

7 8

1.5 **样例输出**

1.17

2.34

1.6 **数据范围与约定**

对于20％的数据，n <= 10, m=0。

对于60％的数据，m=0。

对于100％的数据，n <= 100000, m<=1000，1<=T<=10。

2 **分组** (group.c/cpp/pas)

2.1 **题目描述**

给定一行n个非负整数a[1]..a[n]。现在你可以选择其中若干个数，但不能有超过k个连续的数字被选择。你的任务是使得选出的数字的和最大。

2.2 **输入格式**

第一行两个整数n，k，如题目描述。

接下来一行 n 个数，表示这个序列。

2.3 **输出格式**

输出一行一个数，表示最大的和。

2.4 **样例输入**

5 2

1 2 3 4 5

2.5 **样例输出**

12

2.6 **数据范围与约定**

对于20％的数据，保证1 <= n <= 10。

对于40％的数据，保证1 <= n <= 200。

对于60％的数据，保证1 <= n <= 100000。

对于100％的数据，保证1 <= n <= 2000000，1<=K<=n，0<=a[i]<=100000。

3 **立方**(cube.c/cpp/pas)

3.1 **题目描述**

有一个长为 n 的01序列，每个位置有 a[i] 的概率为1，连续的一段长为k的1会给答案带来 k3 的贡献。

求答案的期望值。

3.2 **输入格式**

第一行一个整数n，表示序列长度。

接下来一行 n 个实数，第 i 个数表示 a[i]。

3.3 **输出格式**

输出一行一个整数，表示保留一位小数的答案 。

3.4 **样例输入**

3

0.5 0.5 0.5

3.5 **样例输出**

6.0

3 .6 **数据范围与约定**

对于30％的数据，保证1<=n<=10。

对于100％的数据，保证1<=n<=100000。

4 **最长公共子序列**(lcs.c/cpp/pas)

4.1 **题目描述**

有两个长分别为n, m的字符串，请求出它们的最长公共子序列。

4.2 **输入格式**

第一行两个整数n, m，表示两个字符串的长度。

接下来两行每行一个字符串，分别表示题目所述的两个字符串。

4.3 **输出格式**

输出一行一个整数，表示答案。

4.4 **样例输入**

5 6

abdac

adabac

4.5 **样例输出**

4

4.6 **数据范围与约定**

对于30％的数据，保证1<=n,m<=10。

对于100％的数据，保证1<=n,m<=6666。