|  |  |  |  |
| --- | --- | --- | --- |
| **题目名称** | **余弦** | **续一秒** | **松鼠与坚果** |
| **程序文件名** | cosine | second | squirrel |
| **输入文件名** | cosine.in | second.in | squirrel.in |
| **输出文件名** | cosine.out | second.out | squirrel.out |
| **每个测试点时限** | 2秒 | 0.5秒 | 1秒 |
| **内存限制** | 512 MB | 512 MB | 512MB |
| **测试点数目** | 10 | 10 | 10 |
| **每个测试点分值** | 10 | 10 | 10 |
| **试题类型** | 传统 | 传统 | 传统 |
| **是否有部分分** | 无 | 无 | 无 |

**提交源程序需加后缀**

|  |  |  |  |
| --- | --- | --- | --- |
| **对于Pascal 语言** | cosine.pas | second.pas | squirrel.pas |
| **对于C语言** | cosine.c | second.c | squirrel.c |
| **对于C++ 语言** | cosine.cpp | second.cpp | squirrel.cpp |

**编译命令：**

C++语言：g++.exe %s.cpp -o %s.exe -Wl,--stack=8388608 -O2

C语言：gcc.exe %s.c -o %s.exe -Wl,--stack=8388608 -O2

Pascal语言：ppc386.exe -Sg %s.pas -O2

**余弦**

**(cosine.pas/c/cpp)**

**题目描述**

给定长度为 的实数序列 , 你需要在数列上进行两类操作：

1. 把 中的每个 加上实数 。
2. 求 中 的和。

**输入格式**

输入包含多组数据，最开头的正整数T (T 指明了数据组数。对于每组数据：

第一行2个整数N、M，表示数列长度与操作个数。

第二行N个实数 ，小数点后最多有3位小数。

接下来M行每行形如“” 或 “”。前者表示一个第1类操作，后者表示一个第2类操作。其中 是整数，而是实数，小数点后最多有3位小数。

**输出格式**

对于每组数据，先输出一行 "Case #k:"，其中k是该组数据的编号，从1开始。

对于每组数据中的每个第2类操作，输出一个实数表示答案，保留3位小数。

|  |  |
| --- | --- |
| **样例输入1** | **样例输出1** |
| 3  3 3  0 -2 -6  1 1 2 -7  2 1 2  2 2 3  5 4  4 2 4 -0 -4  1 1 4 2  2 4 5  1 3 5 5  2 2 3  7 3  0.123 -19.002 -57.507 34.434 -80.886 -17.115 69.843  2 2 7  1 2 7 16.341  2 1 3 | Case #1:  -0.157  0.049  Case #2:  -1.070  -0.649  Case #3:  1.854  -0.842 |

**数据范围与约定**

对于20%的数据，1 ≤ N, M ≤ 1000。

对于另外20%的数据，没有1类操作。

对于另外20%的数据，1类操作均在2类操作之前。

对于100%的数据，1 ≤ N, M ≤ 200000。每个测试点中所有数据的N, M之和各不会超过

**续一秒**

**(second.pas/c/cpp)**

**题目描述**

蛤先生是一位著名的科学家，不久前他得到了一种名叫“熵破坏者”的药剂。这种药被保存在 个瓶子中，其中第个瓶子盛有 升药剂。为了让“熵破坏者”发挥魔力，他必须把所有保存在瓶子中的药剂混合在一起。

由于混合药剂是一件危险的工作，蛤先生取出了他珍藏已久的大容器，并打算把药剂以整瓶为单位，全部倒进大容器中。大容器的容量可以看作无穷大。若大容器中已有 升药剂，则再倒入 升药剂后，大容器中剩余药剂的体积是（神奇的混合方法！）。

最初，大容器是空的，蛤先生可以以任意一种顺序把每瓶药剂倒入大容器。最终，若大容器中还剩余 升药剂，则蛤先生就可以使用一次魔法，把自己的生命周期续 秒。请帮蛤先生求出可能混合成的最大的 ，即蛤先生这次最多可以续几秒。

**输入格式**

输入包含多组数据，在数据开头有一个整数T(T5)，表示组数。对于每组数据：

第一行一个整数N，第二行N个整数 。

**输出格式**

对于每组数据，先输出 "Case #k: "，其中k是该组数据的编号，从1开始，然后紧接着输出最大的 。

|  |  |
| --- | --- |
| **样例输入1** | **样例输出1** |
| 3  4  1 2 2 9  1  -1  10  1 3 0 0 0 1 2 7 3 7 | Case #1: 8  Case #2: 1  Case #3: 6 |

**数据范围与约定**

对于20%的数据，1 ≤ N ≤ 10，

对于60%的数据，1 ≤ N ≤ 50，。

对于100%的数据，1 ≤ N ≤ 200, 。

**松鼠与坚果**

**(squirrel.pas/c/cpp)**

**题目描述**

小取酒是小Cat Rainbow的好朋友，它们一起到森林里玩。这片森林里住着一群松鼠。小取酒很喜欢松鼠，所以它对其中三只松鼠的活动习性进行了详细的跟踪观察。

在这三只松鼠中，第一只喜欢吃榛子，第二只喜欢吃松果，第三只喜欢吃栗子。有一天，它们收集了N个篮子的食物，其中每个篮子里恰好装了1个榛子、1个松果和1个栗子，但是不同的篮子里的食物有不同的美味度。

现在它们邀请小取酒把这N个篮子分成三堆，第1只松鼠会吃掉第1堆每个篮子里的榛子，它获得的美味度就是第1堆篮子里美味度最大的榛子的美味度。同样地，第2只松鼠会吃掉第2堆篮子里的松果，获得第2堆篮子里美味度最大的松果的美味度。第3只松鼠会吃掉第3堆篮子里的栗子，获得第3堆篮子里美味度最大的栗子的美味度。

可是，小取酒还没有吃东西！于是它打算采取适当的方式把篮子分成三堆，使得三只松鼠获得的美味度之和最小。当然，它也有可能不分给某些松鼠食物，也就是说空堆是允许的，它甚至可以把这N个篮子全部分给某一只松鼠。没有获得篮子的松鼠获得的美味度是0。

**输入格式**

第一行一个数 N，表示 N个篮子。

接下来 N 行，每行三个数 ai, bi, ci 表示一个篮子里榛子、松果和栗子的美味度。

**输出格式**

一个整数，表示三只松鼠获得的美味度之和的最小值。

**样例输入**

3

1 2 100

100 4 5

1 100 3

**样例输出**

5

**样例说明**

分给第一只松鼠篮子1和3： {(1, 2, 100), (1, 100, 3)}

分给第二只松鼠篮子2： {(100, 4, 5)}

不分给第三只松鼠篮子： {}

美味度之和：1 + 4 + 0 = 5。

**数据规模与约定**

对于20%的数据，1 ≤ n ≤ 1000。

对于另外30%的数据，1 ≤ ai ≤ 10。

对于100%的数据，1 ≤ n ≤ 100000, 1 ≤ ai, bi, ci ≤ 100000000。