NOIP 模拟赛

Ahtuor: Kewth

题目名称	集合均值	聚烷撑乙二醇	技术情报局	肯德基
时间限制	1s	1s	3s	1s
空间限制	512M	512M	2048M	512M
输入文件	mos.in	pag.in	tio.in	kfc.in
输出文件	mos.out	pag.out	tio.out	kfc.out

注意事项：

- 由于有四道题，题目不是很难，但仍有可能时间不够，每道题都有部分分，请注意时间分配。
- 没有子任务绑定，可以尝试用奇怪的做法骗分，因为即使在正式比赛中，骗分也是得分的有效手段之一。
- 编译选项为 `-std=c++11 -O2` 。

集合均值

（时间限制 1s，空间限制 512M ）

MOS = Mean Of Set

题目描述

有两个可重集合 A, B，初始时 A 只包含一个 0，B 是给定的。

执行以下操作：

1. 在 B 中随机选一个数 y，把 y 从 B 移动到 A。
2. 给答案加上 A 的平均值。
3. 若 B 非空，回到步骤 1 。

求最后答案的期望。

输入格式

第一行一个整数 n, m，表示集合 B 可以分成 m 个大小为 n 的部分。

接下来一行 n 个数表示一个部分，每个部分的数相同。

输出格式

一行一个整数表示答案模 998244353 的结果。

样例

样例 1

```
2 1
1 2
```

```
249561090
```

有两种可能，答案的期望为 $\frac{1}{2}(\frac{3}{2}+2)=\frac{7}{4}$。

样例 2

```
3 3
1 2 3
```

```
346216508
```

该样例中集合 B 为 $\{1,2,3,1,2,3,1,2,3\}$。

数据范围

对于 40% 的数据：$n = 1$ 。
对于 70% 的数据：$n \times m \le 10^5$ 。
对于所有数据：$n \times m \le 2 \times 10^7$ ，$1 \le n \le 10^5, 1 \le m \le 2 \times 10^7$ ，集合中的数是 $[0, 10^9]$ 内的整数。

聚烷撑乙二醇

（时间限制 1s，空间限制 512M）

PAG = Play A Game

题目描述

鲁迅曾经说过，要多去尝试，才能最终发现最优的答案。

鲁迅也还说过，要珍惜当下，把握住眼前的机会。

有 n 个随机数生成器，第 i 个生成器可以均匀随机地生成 $[L_i, R_i]$ 内的一个实数。

现在你要玩个游戏，从第 1 个生成器到第 n 个生成器，每次当前生成器会生成一个数，你需要选择：

1. 相信鲁迅，拿走这个数，游戏结束。
2. 相信鲁迅，放弃这个数和这个生成器，使用下一个生成器（前提是下一个生成器必须存在）。

求使用使得期望答案最大的策略时，期望答案是多少。

输入格式

第一行一个整数 n。

接下来 n 行每行两个非负整数表示 L_i, R_i。

输出格式

一行一个保留恰好 5 位小数（四舍五入）的浮点数表示答案。

样例

样例 1

```
2
0 2
1 1
```

```
1.25000
```

如果第一个生成器生成了比 1 小的数就使用下一个，否则拿走，答案的期望为 $\frac{5}{4}$。

样例 2

```
3
0 9
0 9
0 9
```

```
6.25781
```

数据范围

对于 30% 的数据：$L_i = R_i$ 。
对于另外 40% 的数据：所有 L_i 都为 0 且所有 R_i 都相等。
对于所有数据：$1 \le n \le 10^6, 0 \le L_i \le R_i \le 10^9$ 。

技术情报局

（时间限制 3s，空间限制 2048M ）

TIO = Two In One

题目描述

有这样一道简单题：给定一个序列求所有区间的最大值的和。

还有这样一道简单题：给定一个序列求所有区间的乘积的和。

众所周知：简单题 + 简单题 = 简单题。

所以，给定一个长为 n 的正整数序列，定义一个区间的权值是该区间的数的乘积乘上该区间的最大值，求该序列所有区间的权值和，答案对 p 取模。

输入格式

为了避免输入文件过大，输入仅有一行五个正整数：n, s, l, r, p。

然后通过调用如下函数 `get(n, s, l, r)` 得到一个长为 n 的序列，序列的每个数都是 $[l, r]$ 范围内的整数：

```
#include <vector>

namespace GenHelper {
    unsigned z1, z2, z3, z4, b;
    unsigned rand_() {
        b = ((z1 << 6) ^ z1) >> 13;
        z1 = ((z1 & 4294967294U) << 18) ^ b;
        b = ((z2 << 2) ^ z2) >> 27;
        z2 = ((z2 & 4294967288U) << 2) ^ b;
        b = ((z3 << 13) ^ z3) >> 21;
        z3 = ((z3 & 4294967280U) << 7) ^ b;
        b = ((z4 << 3) ^ z4) >> 12;
        z4 = ((z4 & 4294967168U) << 13) ^ b;
        return (z1 ^ z2 ^ z3 ^ z4);
    }
} // namespace GenHelper

std::vector<int> get (int n, unsigned s, int l, int r) {
    std::vector<int> a;
    using namespace GenHelper;
    z1 = s;
    z2 = unsigned((~s) ^ 0x233333333U);
    z3 = unsigned(s ^ 0x1234598766U);
    z4 = (~s) + 51;
    for (int i = 1; i <= n; i ++) {
        int x = rand_() & 32767;
        int y = rand_() & 32767;
        a.push_back(l + (x * 32768 + y) % (r - l + 1));
    }
    return a;
}
```

输出格式

一行一个正整数表示答案模 p 的结果。

样例

样例 1

```
3 233 1 5 1000
```

```
256
```

该样例的序列为 $\{3,2,5\}$ 。

样例 2

```
1000 2333 1 1000 998244353
```

```
627435628
```

样例 3

```
10000000 666 1 1000 998244353
```

```
123415035
```

数据范围

对于 20% 的数据：$n \le 500$ 。
对于 40% 的数据：$n \le 5000$ 。
对于 60% 的数据：$n \le 10^6$ 。
对于另外 20% 的数据：$l = r$ 。
对于所有数据：$1 \le n \le 10^7, 0 \le s < 2^{30}, 1 \le l \le r \le 10^9, 1 \le p \le 10^9+9$ 。

肯德基

（时间限制 1s，空间限制 512M）

KFC = KeFunction Counting

题目描述

定义 $f(x)=\mu(x)^2x$,求：

$$\sum_{i=1}^{n} f(i)$$

tips: μ 是莫比乌斯函数，$\mu(1)=1$，$x>1$ 时 $\mu(x)$ 有值当且仅当 x 可以表示为 k 个**互不相等**的质数的乘积，此时 $\mu(x)$ 的值为 $(-1)^k$ 。

输入格式

多组数据，第一行一个整数 T 表示数据组数。

接下来 T 行每行一个整数 n 。

输出格式

T 行，每行表示该组数据的答案，鉴于答案可能很大，对 2^{64} 取模。

样例

```
3
10
100
1000
```

```
34
2967
303076
```

数据范围

数据点百分比	$T=$	$1\le n\le$
10	1	5000
10	100	5000
10	100	10^7
20	100	10^9
20	1	10^{12}
10	1	10^{14}
20	100	10^{14}