

扑克牌模拟赛

题目名称	J	Q	K	A
英文名称	jack	queen	king	ace
输入文件	jack.in	queen.in	king.in	ace.in
输出文件	jack.out	queen.out	king.out	ace.out
源文件	jack.cpp	queen.cpp	king.cpp	ace.cpp
时间限制	1s	1s	1s	1s
空间限制	512MiB	512MiB	512MiB	512MiB
子任务数目	10	10	20	20
测试点是否等分	是	是	是	是
题目类型	传统题	传统题	传统题	传统题

注意事项:

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。

J

题目背景



题目描述

你是一个魔术师。你现在要给观众表演魔术。

观众给了你两个整数 n, k ，你要将 $1 \sim 2^n$ 划分为两个集合 a, b ，使得对任意整数 $p \in [0, k]$ ，满足 $\sum_{v \in a} v^p = \sum_{v \in b} v^p$

注意： a 和 b 的交集为空，并集为 $1 \sim 2^n$ 。

输入格式

一行两个整数 n, k 。

输出格式

第一行一个整数 0 或 1，0 表示无解，1 表示有解。

若第一行输出为 1，则第二行第一个整数 k_1 表示集合 a 的大小，接下来 k_1 个整数表示集合 a 中的元素；第三行第一个整数 k_2 表示集合 b 的大小，接下来 k_2 个整数表示集合 b 中的元素。你需要保证 $k_1 + k_2 = 2^n$ 。若有多组解，输出任意一组即可。

若第一行输出为 0，则无需输出其他内容。

样例一

input

```
3 1
```

output

```
1
4 1 4 5 8
4 2 3 6 7
```

限制与约定

对于 30% 的数据, $k \leq 1$ 。

对于另外 20% 的数据, $n \leq 5$ 。

对于 100% 的数据, $1 \leq n \leq 16$, $0 \leq k < n$ 。

Q

题目背景



题目描述

你是一位皇后。你有 n 个皮蛋，每个皮蛋有一个价值 a_i 。

两个皮蛋之间会产生一些不好的味道，第 i 个皮蛋和第 j 个皮蛋放在一起将会产生 $\min\{a_i^{a_j} \bmod 1000000007, a_j^{a_i} \bmod 1000000007\}$ 的不美味度。

你要做两道菜，每道菜里都要放一些皮蛋，一个皮蛋只能且必须放在一道菜中。

一道菜的不美味度为放入这道菜里的所有皮蛋中，不美味度最大的两个皮蛋的不美味度。如果只放了一个皮蛋，则不美味度为 0。

两道菜的总不美味度为两道菜的不美味度中较大的那个。

你现在想让两道菜的总不美味度尽可能小。

输入格式

第一行一个整数 n ，表示皮蛋个数。

第二行 n 个整数 a_1, a_2, \dots, a_n ，表示每个皮蛋的价值。

输出格式

一行一个整数表示答案。

样例一

input

```
4
1 2 1025 1024
```

output

```
1048576
```

explanation

把第一个皮蛋和第三个皮蛋放在第一道菜里，把第二个皮蛋和第四个皮蛋放在第二道菜里。可以证明这样得到的两道菜的总不美味度最小。

样例二

见下发文件中的 `queen/ex_queen2.in` 以及 `queen/ex_queen2.ans`。

样例三

见下发文件中的 `queen/ex_queen3.in` 以及 `queen/ex_queen3.ans`。

限制与约定

对于 10% 的数据， $n \leq 10$ 。

对于 40% 的数据， $n \leq 50$ 。

对于 70% 的数据， $n \leq 300$ 。

对于 100% 的数据， $2 \leq n \leq 2000$ 。

K

题目背景



题目描述

你是一位国王。

你有 n 个整数 a_1, a_2, \dots, a_n 。

你每次可以将一个整数加一，但要花费一块钱。

你希望花一些钱让这 n 个整数的异或和为 0。

你不希望浪费钱财，你希望知道最少要花多少钱。

输入格式

第一行一个整数 T ，表示数据组数。

每组数据的第一行为一个整数 n 。

每组数据的第二行为 n 个整数 a_1, a_2, \dots, a_n 。

输出格式

对每组数据，输出一行一个整数表示答案。

样例一

input

```
4
3
3 4 5
4
5 3 0 7
3
2 3 1
5
7 7 7 7 7
```

output

```
2
1
0
9
```

样例二

见下发文件中的 `king/ex_king2.in` 以及 `king/ex_king2.ans`。

样例三

见下发文件中的 `king/ex_king3.in` 以及 `king/ex_king3.ans`。

限制与约定

对于 100% 的数据, $1 \leq T \leq 100$, $2 \leq n$, $\sum n \leq 10^6$, $0 \leq a_i < 2^{60}$ 。

测试点编号	$n \leq$	$\sum n \leq$	$a_i <$
1	5	500	16
2	5	500	16
3	100	10^3	2^7
4	100	10^3	2^7
5	100	10^3	2^7
6	100	10^3	2^7
7	7	100	2^{60}
8	7	100	2^{60}
9	13	100	2^{60}
10	13	100	2^{60}
11	10^3	10^4	2^{30}
12	10^3	10^4	2^{30}
13	10^3	10^4	2^{30}
14	10^3	10^4	2^{30}
15	10^3	10^4	2^{30}
16	10^6	10^6	2^{30}
17	10^6	10^6	2^{30}
18	10^6	10^6	2^{30}
19	10^6	10^6	2^{60}
20	10^6	10^6	2^{60}

A

题目背景



题目描述

你是 ACE。

你有一个整数数列 a_1, a_2, \dots, a_n 。

你需要解决 m 个询问，每个询问会给定 l, r ，你需要回答以下问题：

对于下标在 $l \sim r$ 之间的**子序列**（共 2^{r-l+1} 个），写出所有子序列的乘积（空序列为 1），这些乘积中，最小的没有出现的正整数是多少。

一个数列的子序列是删除若干元素后得到的数列。

输入格式

第一行一个整数 T 表示数据组数。

每组数据的第一行为两个正整数 n, m 。

第二行为 n 个正整数 a_1, a_2, \dots, a_n 。

接下来 m 行，每行有两个整数 l, r ，表示询问。

输出格式

对于每组数据，输出 m 行，每行一个整数表示答案。

样例一

input

```
1
5 3
2 3 2 5 4
3 5
2 3
1 4
```

output

```
3
4
7
```

样例二

见下发文件中的 `ace/ex_ace2.in` 以及 `ace/ex_ace2.ans`。

样例三

见下发文件中的 `ace/ex_ace3.in` 以及 `ace/ex_ace3.ans`。

限制与约定

对于 100% 的数据, $1 \leq T \leq 5$, $1 \leq n, m \leq 10^5$, $1 \leq a_i \leq 10^9$, $1 \leq l \leq r \leq n$ 。

测试点编号	$n, m \leq$	特殊限制
1	20	
2	100	
3	100	
4	10^3	
5	10^3	
6	10^3	
7	2×10^4	
8	2×10^4	
9	2×10^4	
10	2×10^4	
11	2×10^4	
12	10^5	a 是 $1 \sim n$ 的排列
13	10^5	a 是 $1 \sim n$ 的排列
14	10^5	$a_i \leq 10^3$
15	10^5	$a_i \leq 10^3$
16	10^5	
17	10^5	
18	10^5	
19	10^5	
20	10^5	