

新年模拟赛

whzzt

Jan 1, 2019

题目名称	一个数	一排数	几排数
输入文件名	number.in	array.in	matrix.in
输出文件名	number.out	array.out	matrix.out
时间限制	20s	1s	3s
是否捆绑测试	是	是	否
内存限制	1GB	1GB	1GB
是否有部分分	否	否	是
题目类型	传统	传统	交互
编译开关	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11

注意：1. 评测时的栈空间大小不做单独限制，但使用的总空间大小不能超过内存限制。

2. 考试之中不要交流，AK 之后不要大声喧哗，没有 AK 的同学请安静做题。

3. 不要倦生，不要高喊“这题太难我不会”之类的话。

4. 代码长度限制为 64 KB。

Problem A. 一个数 (number.c/cpp/pas)

Input file: number.in
Output file: number.out
Time limit: 20 seconds
Memory limit: 1 gigabytes

老虎和蒜头是好朋友。

老虎最近在研究整数分解问题，并学习了整数分解中著名的 Pollard Rho 算法。众所周知，Pollard Rho 算法的时间复杂度是 $O(n^{1/4})$ ，即便是分解 30 位的数也非常吃力，于是老虎向蒜头寻求更为优秀的方法。

蒜头告诉老虎，他已经制造出了全新的拥有 1000 个量子比特的量子计算机，因此可以利用 Shor's Algorithm 在 $O(\text{poly}(\log n))$ 内进行因式分解。为了证明这一点，他会对老虎给出的若干个数 N ，将其分解为 $N = \prod p_i^{e_i} (e_i > 0)$ 。但由于念一串长长的数字太累了，蒜头只会告诉老虎最小的 e_i 是多少。

请帮助老虎，对于给定的数据，给出你的答案，以检验蒜头的答案是否正确。由于老虎并不确定蒜头的量子计算机的可靠性，他只要求你给出的答案正确率不小于 90% 即可。

Input

输入的第一行包括一个正整数 T ，表示数据组数。

输入包括 T 组数据，对每组数据包括一行一个大于 1 的整数 N ，表示老虎给出的一个数。

Output

输出 T 行表示答案。

Examples

number.in	number.out
10	1
2	2
9	2
36	1
54	6
64	1
1000000007	5
3659354496	1
998244359987710471	11
86413802648320402620376583	120
1329227995784915872903807060280344576	

对第 7 组数据，有 $3659354496 = 2^7 \times 3^5 \times 7^6$ ；

对第 9 组数据，有 $86413802648320402620376583 = 11^{11} \times 11^{13}$ ；

对第 10 组数据，有 $1329227995784915872903807060280344576 = 2^{120}$ 。

Montgomery Modular Multiplication

在进行大整数运算时，取模往往成为一个非常困难和耗时的问题。为了选手们的身体健康，我们为你提供了一种对于大整数对 N 取模的方法，你可以参考自身情况决定是否使用该方法。

考虑在 $\text{mod } N$ 意义下进行乘法运算。取 $R = 2^l > N$ ，计算 $N'N - R'R = 1$ ，我们定义一个数 x 的 Montgomery form 为 $xR \text{ mod } N$ ，考虑在 Montgomery form 下进行加减乘运算：

$$aR + bR \equiv (a + b)R \pmod{N}$$

$$aR - bR \equiv (a - b)R \pmod{N}$$

$$aR \times bR \equiv (abR)R \pmod{N}$$

容易发现加法和减法不受影响，而乘法只要支持给出 $x \in [0, NR)$ 计算 $xR' \text{ mod } N$ ，由于 $R = 2^l$ ，因此对 R 取模和整除都是很容易的，注意到：

$$xR' \text{ mod } N = \frac{xR'R \text{ mod } NR}{R} = \frac{x(NN' - 1) \text{ mod } NR}{R} = \frac{(xN' \text{ mod } R)N - x}{R} = \left\lfloor \frac{(xN' \text{ mod } R)N}{R} \right\rfloor - \left\lfloor \frac{x}{R} \right\rfloor$$

我们可以解决刚刚的问题，而将一个数转为 Montgomery form 只要计算 xR^2 进行刚刚的操作即可。

本题在 64 位评测机下评测 (可以使用 `__uint128_t` 等类型)，并且对于每组数据，我们允许你出现至多一组答案错误。下发文件中包括以上算法在 $N < 2^{32}$ 时的一份模板，你可以参照这份代码对以上算法进行理解。

Notes

对于 100% 的数据， $T = 10, 2 \leq N \leq 10^{50}$ 。

子任务编号	子任务分值	$N \leq$
1	11	10^7
2	16	10^{18}
3	22	10^{30}
4	28	10^{36}
5	23	10^{50}

Problem B. 一排数 (array.c/cpp/pas)

Input file: array.in
Output file: array.out
Time limit: 1 seconds
Memory limit: 1 gigabytes

老虎和蒜头是好朋友。

老虎最近得到了长度为 n 的数列 a ，对于一个数列，老虎定义了一个优秀度，其通过如下方式计算：令 $x_1 = a_1, x_i = x_{i-1} \bmod a_i$ ，那么这个优秀度就是 x_n 。

现在老虎想要知道在任意排列 a 的情况下，最大的可能优秀度是多少。

Input

输入的第一行包括一个正整数 n ，表示数列的长度。

接下来一行包括 n 个数，表示数列中的元素。

Output

共一行一个数，表示重排后最大的可能优秀度。

Examples

array.in	array.out
5 5 5 10 17 23	3

将序列重排为 23 10 5 17 5 即可。

Notes

对于 100% 的数据， $1 \leq n, a_i \leq 10^5$ 。下表是更详细的数据范围，表中留空意为与以上数据范围相同。

子任务编号	子任务分值	$n \leq$	$a_i \leq$	特殊性质
1	13	10		
2	41		1000	
3	20			a_i 互不相同
4	26			

Problem C. 几排数 (matrix.c/cpp/pas)

Input file: `matrix.in`

Output file: `matrix.out`

Time limit: 3 seconds

Memory limit: 1 gigabytes

这是一道交互题。

老虎和蒜头是好朋友。

老虎最近得到了一个 $N \times N$ 的 01 矩阵，他和蒜头计划用这个矩阵玩一个游戏：蒜头每次可以询问老虎一个矩阵是否是这个 $N \times N$ 的 01 矩阵的子矩阵，蒜头的目标是找到老虎得到的这个矩阵，你需要帮助蒜头在指定的操作次数内完成该任务。

Interaction Protocol

你的任务是实现一个函数：

```
void findMatrix(int N)
```

其中 N 是老虎的矩阵大小，你可以调用下面的函数进行询问：

```
bool isSubMatrix(std::vector<std::vector<char>> matrix)
```

你必须保证你传入的参数是一个矩阵。

你可以调用以下函数返回答案：

```
void foundMatrix(std::vector<std::vector<char>> matrix)
```

你必须保证你传入的参数是一个 N 行 N 列的矩阵，调用该函数后程序将会终止，该函数应当被调用恰好一次。

你可以参考样例交互库对上述描述进行理解。

Notes

本题共包括 20 个测试点，每个测试点的分值为 5 分，其中第 k 个测试点满足 $N = 3k$ 。

不妨假设你的程序调用函数 `isSubMatrix` 的次数为 T ，若你得到的答案错误或是没有进行回答，那么你将得到零分。否则你的得分将取决于 $M = \frac{T}{N^2}$ 的大小，具体来说，假设 M 满足 $M \leq 60, 30, 20, 10, 5$ 中的 x 个限制，你就可以得到 x 分。

我们保证交互库占用空间大小不会超过 128MB，也就是说选手程序的可用空间至少为 896MB。同时，我们保证你调用的函数的实现和下发的样例交互库中的实现相同。

在下发文件中，我们提供了 `matrix.h`, `grader.cpp`, `matrix.cpp` 三个文件，你最终提交的代码应当包含在 `matrix.cpp` 中，并包含头文件 `matrix.h`。

如果你想要命令行编译你的代码，应当执行：`g++ matrix.cpp grader.cpp -o diff -O2 -std=c++11`。