立方数(cubic)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK定义了一个数叫“立方数”，若一个数可以被写作是一个正整数的3次方，则这个数就是立方数，例如1,8,27就是最小的3个立方数。

现在给定一个数P，LYK想要知道这个数是不是立方数。

当然你有可能随机输出一些莫名其妙的东西来骗分，因此LYK有T次询问~

输入格式(cubic.in)

第一行一个数T，表示有T组数据。

接下来T行，每行一个数P。

输出格式(cubic.out)

输出T行，对于每个数如果是立方数，输出“YES”，否则输出“NO”。

输入样例

3

8

27

28

输出样例

YES

YES

NO

数据范围

对于30%的数据p<=100。

对于60%的数据p<=10^6。

对于100%的数据p<=10^18，T<=100。

立方数2(cubicp)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK定义了一个数叫“立方数”，若一个数可以被写作是一个正整数的3次方，则这个数就是立方数，例如1,8,27就是最小的3个立方数。

LYK还定义了一个数叫“立方差数”，若一个数可以被写作是两个立方数的差，则这个数就是“立方差数”，例如7(8-1),26(27-1),19(27-8)都是立方差数。

现在给定一个数P，LYK想要知道这个数是不是立方差数。

当然你有可能随机输出一些莫名其妙的东西，因此LYK有T次询问~

这个问题可能太难了…… 因此LYK规定P是个质数！

输入格式(cubicp.in)

第一行一个数T，表示有T组数据。

接下来T行，每行一个数P。

输出格式(cubicp.out)

输出T行，对于每个数如果是立方差数，输出“YES”，否则输出“NO”。

输入样例

5

2

3

5

7

11

输出样例

NO

NO

NO

YES

NO

数据范围

对于30%的数据p<=100。

对于60%的数据p<=10^6。

对于100%的数据p<=10^12，T<=100。

猜数字(number)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK在玩猜数字游戏。

总共有n个互不相同的正整数，LYK每次猜一段区间的最小值。形如[li,ri]这段区间的数字的最小值一定等于xi。

我们总能构造出一种方案使得LYK满意。直到…… LYK自己猜的就是矛盾的！

例如LYK猜[1,3]的最小值是2，[1,4]的最小值是3，这显然就是矛盾的。

你需要告诉LYK，它第几次猜数字开始就已经矛盾了。

输入格式(number.in)

第一行两个数n和T，表示有n个数字，LYK猜了T次。  
 接下来T行，每行三个数分别表示li,ri和xi。

输出格式(number.out)

输出一个数表示第几次开始出现矛盾，如果一直没出现矛盾输出T+1。

输入样例

20 4

1 10 7

5 19 7

3 12 8

1 20 1

输出样例

3

数据范围

对于50%的数据n<=8，T<=10。

对于80%的数据n<=1000，T<=1000。

对于100%的数据1<=n,T<=1000000,1<=li<=ri<=n,1<=xi<=n（但并不保证一开始的所有数都是1~n的）。

Hint

建议使用读入优化

inline int read()

{

int x = 0, f = 1;

char ch = getchar();

for(; !isdigit(ch); ch = getchar()) if(ch == '-') f = -1;

for(; isdigit(ch); ch = getchar()) x = (x << 1) + (x << 3) + ch - '0';

return x \* f;

}