# 2020 牛客 NOIP 赛前集训营（第二场）
# 提高级

比赛地址：https://ac.nowcoder.com/acm/contest/7607

| 题目名称 | GCD | 包含 | 前缀 | 移动 |
|---|---|---|---|---|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 每个测试点时限 | C/C++ 1 秒，其他语言 2 秒 | C/C++ 1 秒，其他语言 2 秒 | C/C++ 1 秒，其他语言 2 秒 | C/C++ 1 秒，其他语言 2 秒 |
| 内存限制 | C/C++ 512MB，其他语言 1024MB | C/C++ 256MB，其他语言 512MB | C/C++ 256MB，其他语言 512MB | C/C++ 256MB，其他语言 512MB |
| 子任务数目 | 5 | 10 | 10 | 10 |
| 测试点是否等分 | 是 | 是 | 是 | 是 |

## 注意事项

1、所有参与 NOIP 赛前集训营的选手必须遵守约定的纪律:
（1）比赛账号不能外传。
（2）比赛中不能抄袭代码。
（3）比赛中不能恶意卡评测。
2、报名支付账号即为比赛账号。
3、一旦报名 NOIP 赛前集训营活动，不支持退费，请考虑清楚后报名。
4、本活动解释权归牛客网所有，活动介绍未尽事宜以牛客网官方解释为准。
**欢迎关注“比赛自动姬”公众号，关注更多比赛资讯~**

# GCD

**【题目描述】**

我们定义$f(x) = gcd(x除1之外的所有因子)$

即 $x$ 除 1 外所有因子的 $gcd$

询问从$f(a) + f(a+1) + \cdots ... + f(b)$

**【输入格式】**

输入两个正整数 a b

**【输出格式】**

输出一个正整数表示答案

**【样例 1 输入】**

5 7

**【样例 1 输出】**

13

**【数据范围】**

对于 20% 的数据，$1 < a < b \leq 1000$

对于 40% 的数据，$1 < a < b \leq 106$

对于 100% 的数据，$1 < a < b \leq 107$

# 包含

**【题目描述】**

我们定义$A$“包含”$B$的概念是$A\&B = B$，其中&是位运算中的“按位与”。

现在给出一个集合$Q$，这个集合 $n$ 个正整数，$m$ 次询问。每次询问给出一个数字$x$，请回答集合$Q$中是否有一个数字包含$x$。

**【输入格式】**

第一行输入两个正整数$n, m$，意义如题面所示。

接下来一行输入$n$ 个正整数，描述集合$Q$ 中的数字，其中第 $i$ 个数字为$\mathrm{a_i}$。

然后有 $m$行，每行给出一个正整数 $x$，代表询问。

**【输出格式】**

对于每一个询问，输出 yes 或 no 表示答案。

**【样例 1 输入】**

2 2

3 7

4

9

**【样例 1 输出】**

Yes

No

**【数据范围】**

对于 20% 的数据，满足 $n \le 10^5, m \le 10, x \le \mathrm{a_i} \le 1000$

对于 40% 的数据，满足$n \le 10^5, m \le 10^5, x \le \mathrm{a_i} \le 1000$

对于 $100\%$ 的数据，满足 $1 \le n \le 10^5, 1 \le m \le 10^5, 1 \le x \le \mathrm{a_i} \le 10^6$

# 前缀

**【题目描述】**

牛牛有一个$s$串,$s$串仅由 26 个小写英文字母组成，他现在将$s$串进行了无限次的复制扩展成了一个无线循环串。

例如一开始$s$ = "$abc$"，那么牛牛就会将其变为"$abcabcabc...$"

若某个字符串保留其原本字符出现的顺序，并且按照顺序取出若干个字符。可以不连续，可以不取。

我们称取出的这若干个字符连成的字符串为一个子序列。

若连续取出某个字符串的前$k$个字符，组成一个子串，我们称该字符串为原串长度为$k$的前缀。

对于一个字符串$t$，若某字符串的至少一个子序列为$t$。则称它是一个“含$t$序列串”

牛牛想要知道对于给定的$t$，他想要知道$s$的一个最短前缀满足它是一个“含$t$序列串”，它的长度有多长?

由于答案可能非常大，所以他要求你输出答案对 998244353 取余数后的结果即可。

特别的，如果$S$串不存在任何一个前缀满足他是一个“含$t$序列串”，请输出"-1"表示无解。

$t$串中除了 26 个英文字母以外还会出现"*"，表示一个通配符。统配符可以视为任意字母。

例如循环$s$串为"$abcabcabcabc...$"，$t$串为"$a*ca$"时，最短含$t$序列前缀长 4。而当$t$串为"$a**ca$"时，最短含$t$序列前缀长 7。

除此之外，牛牛输入的$t$串还可能非常非常长，最长可以达到 10^{10^5}这么长。

所以他想了一种压缩方法，来快速读入 t 串。

具体来说，它输入的 t 串中除了"*"和 26 个小写英文字母以外，还会跟有一些正整数。

在读入字符串时，这些数字表示它前面字母或者"*"重复的次数。

例如 a5bc*3，表示"aaaaabc***"。输入的正整数不含前导 0。

**【输入格式】**

第一行输入一个仅包含 26 个小写英文字母的字符串$s$

第二行输入一个正整数$n$表示，$t$串的数目。

接下来输入$n$行

再输入一行一个字符串$t$，表示压缩后的查询串。查询串仅包含 26 个小写英文字母，星号'*'，以及数字。

**【输出格式】**

对于每一个查询，如果至少存在一个 s 的前缀满足“最短含 t 序列串”的定义，请输出 s 的最短含 t 序列前缀的长度对 998244353 取余数后的结果。

否则请输出"-1"表示无解。

**【样例 1 输入】**

abc

3

a*ca

a**ca

a*2ca

**【样例 1 输出】**

4

7

7

**【样例 1 解释】**

S 串为:abcabcabcabcabc....

包含 a*ca 作为子序列的最短前缀为 abca。

包含 a**ca 作为子序列的最短前缀为 abcabca。

a*2ca 表述的 T 串和 a**ca 等价。

**【样例 2 输入】**

nowcoder

2

o10000000000000000000000000000000000000000000000000000000000000000

winterzz1

**【样例 2 输出】**

110162207

-1

**【样例 2 说明】**

最短含 t 前缀长度为 399999999999999999999999999999999999999999999999999999999997，该数对 998244353 取余数的结果为 110162207

**【数据范围】**

对于前 10%的测试数据保证$1 \le n \le 100, 1 \le |s|, |t| \le 10$且 t 串中不包含数字以及'*'。

对于前 20%的测试数据保证$1 \le n \le 100, 1 \le |s|, |t| \le 10$且 t 串中不包含数字。

对于前 30%的测试数据保证$1 \le n \le 1000, 1 \le |s|, |t| \le 1000$且 t 串中不包含数字。

对于前 60%的测试数据保证$1 \le n \le 10^5, 1 \le |s| \le 10^4, 1 \le |t| \le 10^5$且 t 串中的数字值域范围在$[1, 10^9]$内。

对于前 100%的测试数据保证$1 \le n \le 105, 1 \le |s| \le 104, 1 \le |t| \le 105, \sum |t| \le 10^6$且 t 串中的数字值域范围在 10^{10^5}内。

注意，|t|仅表示输入时的压缩串的长度，不代表解压缩后的长度，解压后 t 串的长度最长可以达到 10^{10^5}。

# 移动

**【题目描述】**

牛牛被困在了一个房间里，他可以看到房间的出口，但是想要到达出口，需要经过 $n$ 道闸门。我们可以根据这些闸门离牛牛的距离进行编号，离牛牛最近的闸门记为 1 号闸门，离牛牛最远的记为$n$号闸门。

牛牛每秒都可以选择前进到下一闸门，后退到上一闸门，或者原地不动。(从起点到第一道闸门，从第$n$道闸门到出口的时间也是一秒)

这些闸门在一些时刻是关闭的，无法通行，剩下的时刻是开启的，可以通行。

注意：如果牛牛所在的位置有一个闸门即将关闭，他在此时选择原地不动，就会被闸门夹到，变成牛排。牛牛想在不变成牛排的前提下走到出口，他想知道最短需要多少秒才能走到出口，如果他永远无法走到出口，输出-1。

在每一秒内，首先牛牛进行移动，然后闸门进行开/关的动作。

**【输入格式】**

第一行给出 $N, M$，分别代表有$N$道闸门，$M$个信息

接下来有 $M$ 行，每行代表一个闸门关闭的信息，包含三个数字 $a\ b\ c$，代表第 a 道闸门会在 $[b, c]$的时间内关闭

保证这些信息之间不相交。(即不会有某一道闸门在$[b_1, c_1]$和$[b_2, c_2]$的时间关闭，且 $, b_1 < b_2 < c_1$ 。

**【输出格式】**

输出一行一个整数代表牛牛到达出口所需要的最短时间

**【样例 1 输入】**

4 2

1 2 50

3 2 5

**【样例 1 输出】**

8

**【样例 1 说明】**

牛牛在 1 秒时到达 1 号闸门，2 秒时到达 2 号闸门，等到 6 秒时到达 3 号闸门，7 秒到达 4 号闸门，8 秒到达出口

**【数据范围】**

20% 的数据，满足 $1 \leq N, M \leq 20$

60% 的数据，满足 $1 \leq N, M \leq 2000$

100% 的数据，满足 $1 \leq N, M \leq 100000, 1 \leq a \leq N, 1 \leq b \leq c \leq 10^9$