

题目名称	徐老师的假期奖金	徐老师的装备强化	徐老师的窗外风景	徐老师的弱循环节
文件名	money.cpp	equip.cpp	happy.cpp	cycle.cpp
输入文件	money.in	equip.in	happy.in	cycle.in
输出文件	money.out	equip.out	happy.out	cycle.out
时间限制	1000ms	1000ms	1500ms	2000ms
空间限制	512MB	512MB	512MB	512MB

徐老师的假期奖金

文件读写

- 输入文件money.in
- 输出文件money.out

题目限制

- 1000ms
- 512MB

题目描述

马上就要放暑假啦！

为了激励一下公司的员工，徐老师决定给所有人发奖金！

发奖金的方式非常简单，徐老师会在桌上一字排开 n 张纸团，每张纸团上会写着一个数字 a_i ，为了平衡公司成本，这里的数字可能会存在负数

游戏规则是：每人只能抓一次，只能抓取一段连续的纸团，所有纸团上的数字和就是这次可以获得的奖金，如果有人不幸获得了负数的奖金，那么就意味着他要被扣掉对应的工资，所以徐老师也允许大家选择不抓纸团，那么奖金就是 0 元

现在石老师是第一个抽奖金的人，石老师想知道，他一共有多少种不同的方案可以获得至少 k 元的奖金？

输入格式

输入第一行包含一个整数 n, k 含义如题

输入第二行包含 n 个整数， a_i 依次表示每个纸团上的数字

输出格式

输出一个整数表示有多少种不同的方案

数据范围

对于 30% 的数据: $1 \leq n \leq 2000$, $-10^8 \leq k \leq 10^8$, $-10^4 \leq a_i \leq 10^4$ 。

对于另外 10% 的数据: $a_i < 0, k = 0$

对于 100% 的数据: $1 \leq n \leq 3 \times 10^5$, $-10^{18} \leq k \leq 10^{18}$, $-10^9 \leq a_i \leq 10^9$ 。

样例输入1

```
4 1
-5 4 3 -3
```

样例输出1

```
5
```

样例解释1

五种方案分别为 $[-5, 4, 3], [4], [4, 3], [4, 3, -3], [3]$

样例输入2

```
10 -1
-5 4 5 -4 1 3 5 8 -4 5
```

样例输出2

```
52
```

样例输入3

```
5 0
-1 -1 -1 -1 -1
```

样例输出3

```
1
```

文件读写

- 输入文件equip.in
- 输出文件equip.out

题目限制

- 1000ms
- 512MB



题目描述

众所周知，徐老师很喜欢玩游戏，这天他玩的游戏中新出了一个强(ke)化(jin)系统，允许玩家在一件装备上镶嵌强化石

而常规的概率强化已经无法吸引玩家了，所以新推出的这个强化系统的强化方式是这样的

每件装备拥有一个独立的强化界面，这个强化界面可以看作一个 $n * m$ 的矩阵，每个点都可以镶嵌一颗强化石

在镶嵌了强化石后，装备的强化系数就是强化石构成的等腰三角形个数

这里的等腰三角形是指（镶嵌界面中  表示为空， 表示镶嵌了强化石）
大小为 1 的三角形

```
*

```

大小为 2 的三角形

```
. *.
***

```

大小为 3 的三角形

```
. . *.
. ***.
*****

```

依次类推...

比如一件装备拥有一个 $3 * 5$ 的强化界面

```
. . *.
. ****
*****

```

上述装备镶嵌了 10 颗强化石，获得的强化系数是 15

现在徐老师已经给自己的装备镶嵌好了强化石，他想知道这件装备的强化系数是几？

输入格式

输入第一行包含两个正整数 n, m 表示这件装备的强化界面

接下来 n 行，每行一个长度为 m 的字符串，仅包含 `.` 和 `*`，其中 `.` 表示为空，`*` 表示这个格子镶嵌了强化石

输出格式

输出一个整数表示强化系数

数据范围

对于 30% 的数据： $n, m \leq 5$

对于 60% 的数据： $n, m \leq 500$

对于 100% 的数据： $n, m \leq 1000$

样例输入

```
3 5
..*..
.****
*****
```

样例输出

```
15
```

徐老师的窗外风景

文件读写

- 输入文件happy.in
- 输出文件happy.out

题目限制

- 1500ms
- 512MB

题目描述

终于放假了的徐老师又双叒来到了一个新的城市旅游！

这一次他来的城市是一座以 `窗外风景` 为看点的城市，也就是说在这个城市旅游，重点在于路上的风景，而不是具体的某个景点

这个城市一共有 n 个街区，街区之间由 $n - 1$ 条道路连接，每条道路的风景美丽程度可以形式化的用一个数字来表示

也就是说第 i 条道路会连接编号为 u_i, v_i 的两个街区，并且这条道路的风景美丽程度为 a_i

但是这个城市的景点之间相似度比较高，所以会导致在旅游时，去过的景点越多，反而会因为审美疲劳而觉得景点没那么有意思，而徐老师通过一些道路从一个街区 x 到达街区 y 时，他这次旅行最终获得的快乐度将会是经过的每条道路的风景美丽程度的最大公约数

也就是说如果徐老师如果连续通过三条道路 i, j, k ，那么他最终会获得的快乐度将是 $\gcd(a_i, a_j, a_k)$

现在徐老师正在做旅行规划，但是他又是一个很纠结的人，他这次旅游想要经过正好 m 条道路，但是又怕经过的道路太多导致快乐度下降

所以他决定问问你，当徐老师决定正好经过 m 条道路（不允许重复经过同一条道路）的情况下，如何规划路径可以使得他获得的快乐度是最高的？

输入格式

输入第一行包含一个整数 n ，表示街区数量

接下来 $n - 1$ 行，每行三个整数 u_i, v_i, a_i 表示一条道路信息

接下来一行输入一个整数 T ，表示 T 次询问

接下来 T 行，每行一个整数 m ，表示徐老师想询问正好经过 m 条道路的情况下，他能获得的最高快乐度是多少

输出格式

对于每次询问输出一个整数，表示答案，如果路径不存在，则输出 -1

数据范围

对于 10% 的数据满足： $a_i = 1$

对于另外 20% 的数据满足： $1 \leq n \leq 1000$

对于另外 30% 的数据满足： $1 \leq a_i \leq 100$

对于 100% 的数据满足：

$1 \leq n \leq 400000, 1 \leq T \leq \min(100000, n), 1 \leq u_i, v_i, m \leq n, 1 \leq a_i \leq 1000000$ ，保证题目数据随机生成

样例输入

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

样例输出

```
-1
8
2
```

样例解释

不存在能够经过 4 条道路的方案，所以答案为 -1

当经过 1 条道路时，选择 4 - 5 这条道路，答案就是 8

当经过 3 条道路时，由于必须经过 1 - 2 这条道路，所以答案只能是 $\gcd(2, ?, ?)$ 必然为 2

徐老师的弱循环节

文件读写

- 输入文件cycle.in
- 输出文件cycle.out

题目限制

- 2000ms
- 512MB

题目描述

徐老师最近在学习 *Hash* 算法，*Hash* 算法有一个经典的应用就是进行字符串匹配求一个字符串的循环节长度。

于是徐老师准备自己生成一些字符串当做测试数据，测试代码是否正确

他会先自己随机写一个仅包含大小写字母的字符串 S

然后用以下代码依次生成一个字符串序列 $A_0 \sim A_n$

```
int n = S.size();
A[0] = "";
for (int i = 1; i <= n; ++i){
    if (S[i - 1] >= 'A' && S[i - 1] <= 'Z'){
        A[i] = A[i - 1] + char(S[i - 1] + 32);
    } else {
        A[i] = S[i - 1] + A[i - 1];
    }
}
```

但是徐老师觉得求循环节太简单了，于是他设计了一个新的概念——弱循环节

对于一个字符串 B 来说，一个整数 len 如果满足对于任意 $B_i = B_{i+len} (1 \leq i \leq |B| - len)$ ，那么就认为 len 是 B 的一个弱循环节，一个字符串可能有多个弱循环节

例如 $ABCAB$ 的弱循环节有两个，分别是 3 和 5

单纯求字符串的弱循环节也还是过于简单了

于是徐老师会给出一个序列 $a_1 \sim a_m$ ，并给出 T 次询问

第 i 次询问徐老师会给出三个整数 l_i, r_i, pos_i ，你的任务是在 a 序列中找出一个 a_x 满足 $x \in [l_i, r_i]$ 且 a_x 是 A_{pos_i} 的弱循环节，如果有多个满足条件的 a_x 请输出最小的 a_x （数值最小而不是 x 最小）

输入格式

输入第一行包含一个字符串 S ，保证仅由大小写字母组成

输入第二行包含一个整数 m ，表示序列 a 的长度

输入第三行包含 m 个整数，分别表示 a_i

输入第四行包含一个整数 T ，表示询问次数

接下来 T 行，每行包含三个整数 pos_i, l_i, r_i ，表示一次询问

输出格式

对于每次询问，输出一个整数，表示第 i 次询问的答案，如果不存在符合条件的 a_x ，则输出 "No answer!"

数据范围

对于 20% 的数据满足 $n, m, T \leq 100$

对于另外 20% 的数据满足 $r_i - l_i \leq 100$

对于另外 20% 的数据满足 a 序列单调不减

对于 100% 的数据满足 $1 \leq n, m, T \leq 500000, 1 \leq a_i, pos_i \leq n, 1 \leq l_i \leq r_i \leq m$

样例输入

```
AABAaba
9
4 3 2 1 7 5 3 6 1
6
1 4 4
2 1 4
2 1 3
3 3 5
5 4 7
7 8 9
```

样例输出

```
1
1
2
No answer!
3
6
```

样例解释

对于第一次询问， $A[1] = a$ ，所以满足条件的弱循环节长度为 1

对于第二次询问， $A[2] = aa$ ，所以满足条件的弱循环节长度为 1，2，最小的是 1

对于第三次询问，满足条件的只有 2，所以答案是 2
