Problem A. 有趣的数

输入文件:	numbers.in
输出文件:	numbers.out
时间限制:	1 second
内存限制:	128 megabytes

KK 闲来无事就自己定义了一类有趣的数。

如果一个正整数满足在 B_1 进制下的位数是 D_1，而且在 B_2 进制下的位数是 D_2，KK 就认为他是一个有趣的数。

现在，KK 想知道满足要求的数一共有多少个。

KK 为了不为难你，他只需要你考虑所有不超过 10^{18} 的数是不是有趣的。换句话说，KK 认为所有大于 10^{18} 的数都是不有趣的。

输入格式

输入数据包含一行，四个整数，分别是 B_1，D_1，B_2，D_2，意义如题面所述。

输出格式

输出一行包含一个整数，表示答案。

样例

numbers.in	numbers.out
3 1 5 1	2
10 2 2 4	6

样例解释

对于第一个样例：满足在 3 进制下一位的正整数有 $1, 2$，这两个数在五进制下也是一位，所以满足要求题意的数有两个。

约束

- $2 \leqslant B_1, B_2 \leqslant 100$。
- $1 \leqslant D_1, D_2 \leqslant 20$。

Problem B. 木棒

输入文件:	sticks.in
输出文件:	sticks.out
时间限制:	4 seconds
内存限制:	128 megabytes

KK 手上有 12 根木棒，它们的长度依次为 $l_1, l_2, \cdots, l_{12}$。他想用一些木棒来拼出三角形。

KK 规定，每一根木棒只能用在至多一个三角形中。他想知道，最多可以组成多少个三角形。

输入格式

输入包含多组测试数据。第一行包含一个整数 T，表示测试数据的组数。随后的内容是各组测试数据。

对于每组测试数据：仅一行，包含十二个整数 $l_1, l_2, \cdots, l_{12}$。

输出格式

对于每组测试数据，输出一行包含一个整数，表示最多能组成的三角形个数。

样例

sticks.in	sticks.out
5 1 2 1 3 1 4 1 5 1 6 1 7 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 5 8 13 21 34 55 89 144 233 2 3 6 15 27 59 72 83 121 159 201 234 2 2 4 8 16 32 64 128 256 512 1024 1281	4 3 0 2 1

约束

- $1 \leqslant T \leqslant 6000$.
- $1 \leqslant l_i \leqslant 10^9$.

子任务

子任务编号	分值	T	l_i
1	10	$= 1$	$\leqslant 10^9$
2	50	$\leqslant 20$	$\leqslant 10^9$
3	40	$\leqslant 6000$	$\leqslant 10^9$

Problem C. 装饰

输入文件:	tree.in
输出文件:	tree.out
时间限制:	1 second
内存限制:	128 megabytes

春节来了，KK 开始装饰家里巨大的银杏树。

KK 把这棵树抽象成一个有根树的形状，这棵树一共有 n 个节点，编号为 $1, 2, \cdots, n$，其中节点 1 是树根，每一个树节点 $x(x > 1)$ 都有一个父亲节点 P_x。显然节点 1 没有父节点。

KK 为了装饰的整体美观程度，他要求每一个节点 x 以及他子树上总的装饰品数量至少为 C_x。

由于每一个树节点都不同，所以在每一个节点上布置装饰品的时间也不尽相同。但 KK 已经知道在节点 x 布置一个装饰品的时间是 T_x，也就是说，如果在节点 x 布置 k 个装饰品所需要的时间是 $k \cdot T_x$。

现在他希望尽可能减少布置装饰品的时间，他想知道完成预想的布置，最少需要多少时间。

输入格式

第一行包含一个整数 n，第 2 至 $n+1$ 行，第 $i+1$ 行包含三个隔开的整数 P_i, C_i, T_i。

输入数据中，节点 1 的父亲输入为 -1。

输出格式

输出一个正整数表示答案。答案保证在 64 位有符号整型范围内。

样例

tree.in	tree.out
3 -1 3 3 1 1 4 1 3 2	10

样例解释

KK 预想的方案有以下要求

节点编号 x	子树所需要最少的装饰品数量 C_i	放一个装饰品的时间 T_i
1	3	3
2	1	4
3	3	2

KK 可以按照下面的方法放置装饰品（你不可能再找到比这个时间更短的方案）：

- 在节点 2 放置要求的 1 个装饰品，花费时间 4。

- 在节点 3 放置要求的 3 个装饰品，花费时间 6。
- 在节点 1 的子树中，放置已经达到要求，可以不放置新的装饰品，花费时间 0。

所以布置的总时间为 10。

约束

- $1 \leqslant P_i \leqslant n \leqslant 10^5$
- $0 \leqslant C_i \leqslant 10^7$.
- $1 \leqslant T_i \leqslant 100$.

子任务

子任务编号	分值	n
1	20	$\leqslant 20$
2	30	$\leqslant 500$
3	20	$\leqslant 8000$
4	30	$\leqslant 10^5$

Problem D. 翻转硬币

输入文件: coin.in
输出文件: coin.out
时间限制: 1 second
内存限制: 1024 megabytes

KK 去参加 Google 面试了。面试官给了 KK 一共 n 个硬币，并把他们排成一列。

现在面试官给了一个整数 m，他要求 KK 通过以下两种操作，使得硬币组成的数列前 $n-m$ 位和后 $n-m$ 位匹配。具体的说，设硬币组成的数列为 $a_1, a_2, \cdots, a_n$，面试官要求 $a_1 = a_{m+1}, a_2 = a_{m+2}, \cdots, a_{n-m} = a_n$。

KK 所选择的操作，必须是下面两种中的其中一种：

- 选择其中一个硬币，并且翻转它。
- 选择前 $k \cdot m$ 位（k 为任意正整数），并且翻转它们。

为了博取面试官的好感，KK 自然想用最少的翻转次数来完成面试官的要求。

输入格式

第一行输入一个仅包含 0/1 的字符串，字符串的长度为 n。

第二行输入一个整数 m，含义如题目所述。

输出格式

输出包含一个整数，表示所需要的最小翻转次数。

样例

coin.in	coin.out
00111000 1	2
101100001101 3	2

样例解释

第一个样例的两个操作分别是：

1）执行操作 2 并且令 $k = 5$，操作后变成 11000000。

2）执行操作 2 并且令 $k = 2$，操作后变成 00000000。

约束

- $1 \leqslant m \leqslant n \leqslant 300$.

子任务

子任务编号	分值	n	m
1	30	$\leqslant 10$	$\leqslant 10$
2	10	$\leqslant 10$	$\leqslant n$
3	20	$\leqslant 300$	$\leqslant 10$
4	40	$\leqslant 300$	$\leqslant n$