

## NOIP 模拟试题

题目名称	程序文件名	输入文件名	输出文件名	时限
混乱的队伍	mixup2.pas/c/cpp	mixup2.in	mixup2.out	1s
安慰员工	cheer.pas/c/cpp	cheer.in	cheer.out	1s
齿轮	baler.pas/c/cpp	baler.in	baler.out	1s

### 混乱的队伍 (mixup2. pas/c/cpp)

#### 【题目描述】

LongDD 的  $N$  ( $4 \leq N \leq 16$ ) 个员工每人都有一个唯一的编号  $S_i$  ( $1 \leq S_i \leq 25,000$ )。员工们为他们的编号感到骄傲，所以每个人都把他的编号刻在一个金牌上，并且把金牌挂在他们的脖子上。

员工们对在吃饭的时候被排成一支“混乱”的队伍非常反感。如果一个队伍里任意两相邻的人的编号相差超过  $K$  ( $1 \leq K \leq 3400$ )，它就被称为是混乱的。比如说，当  $N = 6$ ,  $K = 1$  时, 1, 3, 5, 2, 6, 4 就是一支“混乱”的队伍，但是 1, 3, 6, 5, 2, 4 不是(因为 5 和 6 只相差 1)。

那么，有多少种能够使员工们排成“混乱”的队伍的方案呢？

程序名：mixup2

#### 【输入格式】

- \* 第 1 行：用空格隔开的两个整数  $N$  和  $K$
- \* 第 2.. $N+1$  行：第  $i+1$  行包含了一个用来表示第  $i$  个员工的编号的整数： $S_i$

#### 【输入样例】

```
4 1
3
4
2
1
```

#### 【输出格式】

- \* 第 1 行：只有一个整数，表示有多少种能够使员工们排成“混乱”的队伍的方案。答案保证是一个在 64 位范围内的整数。

#### 【输出样例】

2

【输出解释】

两种方法分别是：

3 1 4 2

2 4 1 3

## 安慰员工(cheer.pas/c/cpp)

### 【题目描述】

LongDD 变得非常懒, 他不想再继续维护供员工之间供通行的道路. 道路被用来连接  $N$  ( $5 \leq N \leq 10,000$ ) 个房子, 房子被连续地编号为  $1..N$ . 每一个房子都是一个员工的家. LongDD 计划除去  $P$  ( $N-1 \leq P \leq 100,000$ ) 条道路中尽可能多的道路, 但是还要保持房子之间的连通性. 你首先要决定那些道路是需要保留的  $N-1$  条道路.

第  $j$  条双向道路连接了房子  $S_j$  和  $E_j$  ( $1 \leq S_j \leq N$ ;  $1 \leq E_j \leq N$ ;  $S_j \neq E_j$ ), 而且走完它需要  $L_j$  ( $0 \leq L_j \leq 1,000$ ) 的时间. **没有两个房子是被一条以上的道路所连接.**

员工们非常伤心, 因为她们的交通系统被削减了. 你需要到每一个员工的住处去安慰她们. 每次你到达第  $i$  个房子的时候(即使你已经到过), 你必须花去  $C_i$  ( $1 \leq C_i \leq 1,000$ ) 的时间和员工交谈.

你需要从某一个房子出发(这是供你选择的), 并最终回到这个房子。期间, 你要经过每个房子至少一次, 并且当你经过某个房子的时候, 你必须和这个房子里的员工交谈(即使你已经到过).

假设 LongDD 采纳了你的建议, 请计算出使所有员工都被安慰的最少时间.

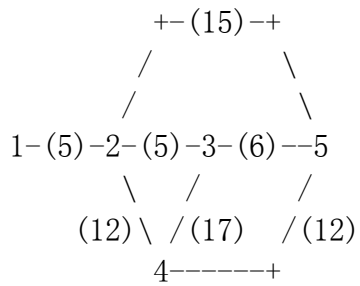
### 【输入格式】

- \* 第 1 行: 用空格隔开的两个整数  $N$  和  $P$
- \* 第  $2..N+1$  行: 第  $i+1$  行包含了一个整数:  $C_i$
- \* 第  $N+2..N+P+1$  行: 第  $N+j+1$  行包含用空格隔开的三个整数:  $S_j$ ,  $E_j$  和  $L_j$

### 【输入样例】

```
5 7
10
10
20
6
30
1 2 5
2 3 5
2 4 12
3 4 17
2 5 15
3 5 6
4 5 12
```

【输入解释】



【输出格式】

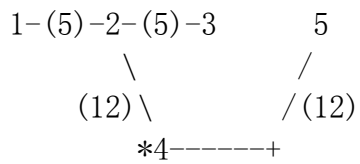
\* 第 1 行：一个整数，所需要的总时间(包含和在你所在的房子的员工的两次谈话时间).

【输出样例】

176

【输出解释】

保持这些路径：



从房子 4 出发, 然后按照 4, 5, 4, 2, 3, 2, 1, 2, 4 的顺序来访问员工们。

路上的总花费是：12+12+12+5+5+5+5+12=68。共和员工 1 交谈了 1 次，和 2 交谈了 3 次，和 3 交谈了 1 次，和 4 交谈了 3 次，和 5 交谈了 1 次。交谈总花费 10\*1+10\*3+20\*1+6\*3+30=108。总共需要 176 个单位的时间。

## 齿轮(baler.pas/c/cpp)

### 【问题描述】

LongDD 新买的机器的内部结构大概算世界上最混乱的了，它不象普通的机器一样有明确的内部传动装置，而是， $N$  ( $2 \leq N \leq 1050$ ) 个齿轮互相作用，每个齿轮都可能驱动着多个齿轮。

LongDD 记录了对于每个齿轮  $i$ ，记录了它的 3 个参数： $X_i, Y_i$  表示齿轮中心的位置坐标 ( $-5000 \leq X_i \leq 5000; -5000 \leq Y_i \leq 5000$ )； $R_i$  表示该齿轮的半径 ( $3 \leq R_i \leq 800$ )。驱动齿轮的位置为  $0, 0$ ，并且 FJ 也知道最终的工作齿轮位于  $X_t, Y_t$ 。

驱动齿轮顺时针转动，转速为 10,000 转/小时。你的任务是，确定传动序列中所有齿轮的转速。传动序列的定义为，能量由驱动齿轮传送到工作齿轮的过程中用到的所有齿轮的集合。对能量传送无意义的齿轮都应当被忽略。在一个半径为  $R_d$ ，转速为  $S$  转/每小时的齿轮的带动下，与它相接的半径为  $R_x$  的齿轮的转速将为  $-S \cdot R_d / R_x$  转/小时。 $S$  前的负号的意思是，一个齿轮带动的另一个齿轮的转向会与它的转向相反。

LongDD 只对整个传动序列中所有齿轮速度的绝对值之和感兴趣，你的任务也就相应转化成求这个值。机器中除了驱动齿轮以外的所有齿轮都被另外某个齿轮带动，并且不会出现 2 个不同的齿轮带动同一个齿轮的情况。

相信你能轻易地写个程序来完成这些计算：)

程序名：baler

### 【输入格式】

- \* 第 1 行：3 个用空格隔开的整数： $N, X_t, Y_t$
- \* 第 2.. $N+1$  行：第  $i+1$  描述了齿轮  $i$  的位置及半径： $X_i, Y_i$ ，以及  $R_i$

### 【输入样例】

```
4 32 54
0 0 10
0 30 20
32 54 20
-40 30 20
```

### 【输入说明】

机器里一共有 4 个齿轮，位于  $0, 0$  的是半径为 10 的驱动齿轮，它带动了位于  $0, 30$  的，半径为 20 的某个齿轮。这个齿轮又间接带动了位于  $32, 54$ ，半径为 20 的工作齿轮，以及一个位于  $-40, 30$ ，半径同样为 20 的冗余的齿轮。

【输出格式】

\* 第 1 行：输出所有在传动中起到作用的齿轮转速的绝对值，包括驱动齿轮和工作齿轮。只需要输出答案的整数部分

【输出样例】

20000

【输出说明】

齿轮	位置	半径	转速
1	(0, 0)	10	10, 000
2	(0, 30)	20	-5, 000
3	(32, 54)	20	5, 000

-----  
齿轮转速绝对值之和:20, 000