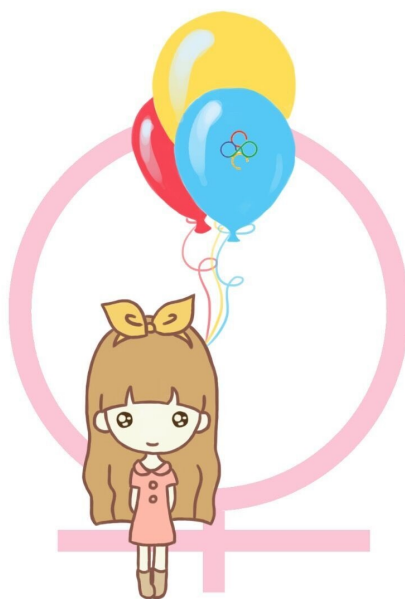


# 2021 年中国大学生程序设计竞赛 女生专场



山东, 淄博

2021 年 10 月 31 日

## Problem A. 公交线路

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

比特镇只有一条双向的公交线路，这条线路一共经过  $n$  个不同的站点，从头到尾依次编号为  $1, 2, \dots, n$ ，其中第  $i$  个站点的名称包含  $k_i$  个汉字。由于是双向道路，你既可以从 1 号站点坐车前往  $n$  号站点，也可以从  $n$  号站点坐车前往 1 号站点，但是从 1 号站出发的车到达  $n$  号站点之后不会折返，从  $n$  号站出发的车到达 1 号站点之后也不会折返，必须要下车换乘。在行驶到两个站点中间时，公交会播报沿着行驶方向即将到达的站点名称。

小 Q 的公司位于  $x$  号站点，小 Q 的家则位于  $y$  号站点。现在小 Q 已经从  $x$  号站点上车准备回家，但是车上实在太多人了，小 Q 上车之后并不能通过站牌看到现在到了哪一站，也听不清车上即将到站的播报，只能听到播报的站点名称包含多少个汉字。

因为粗心的小 Q 经常坐反公交，在车上播报了  $m$  次即将到达的站点名称后，小 Q 想立即确认自己有没有坐反方向。如果真的坐反了，公交到站停靠后小 Q 就要立即下车换乘到反方向的车上。小 Q 非常确信如果自己坐对了方向是不会坐过站的，也就是说播报的站点数一定不大于  $|x - y|$ 。

### Input

第一行包含三个正整数  $n, x, y$  ( $2 \leq n \leq 10, 1 \leq x, y \leq n, x \neq y$ )，依次表示站点的数量、公司所在的站点以及家所在的站点。

第二行包含  $n$  个正整数  $k_1, k_2, \dots, k_n$  ( $1 \leq k_i \leq 9$ )，依次表示每个站点的字数。

第三行包含一个正整数  $m$  ( $1 \leq m \leq |x - y|$ )，表示沿途经过的站点数。

第四行包含  $m$  个正整数  $p_1, p_2, \dots, p_m$  ( $1 \leq p_i \leq 9$ )，依次表示每次播报的字数，保证小 Q 听到的  $m$  次即将到站的播报是合法的。

### Output

如果通过  $m$  次即将到站的播报能确定小 Q 一定没有坐反方向，请输出“Right”；如果能确定小 Q 一定坐反方向了，请输出“Wrong”；如果通过  $m$  次即将到站的播报不能确定乘坐方向，也就是说小 Q 既可能坐对方向，也可能坐反方向，请输出“Unsure”。

## Examples

stdin	stdout
7 4 6 3 2 3 4 5 4 3 2 5 4	Right
7 4 6 3 2 3 4 5 4 3 2 3 2	Wrong
7 4 6 3 3 3 3 3 3 3 2 3 3	Unsure

## Problem B. 攻防演练

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

小 Q 和小 C 在比特公司的系统中进行攻防演练, 这个系统经过特殊设定, 只能接收任何只含有前  $m$  个小写英文字母的非空字符串作为输入命令。

小 Q 事先准备了一个长为  $n$  的字符串  $s = s_1s_2 \dots s_n$ , 为了能够在演练时输入到系统中, 这个字符串只会包含前  $m$  个小写英文字母。攻防演练一共进行  $q$  轮, 每轮开始时小 Q 会选择一个  $s$  的非空子串  $s_{l,r} = s_l s_{l+1} \dots s_r$  输入到系统中作为本轮演练的防火墙规则。当防火墙规则配置完成之后, 对于任何输入到系统中的命令  $c$ , 只要  $c$  是  $s_{l,r}$  的子序列就会被防火墙拦截。

小 C 的任务是在每轮演练中, 在小 Q 完成本轮防火墙规则的配置之后, 输入一个不被防火墙拦截的命令。为了节约时间, 小 C 想知道每轮演练需要输入的最短字符串的长度是多少。也就是说, 小 C 想找到最小的正整数  $k$ , 存在一个长为  $k$  的字符串  $t = t_1 t_2 \dots t_k$ , 使得不存在任意一个长为  $k$  的序列  $p_1, p_2, \dots, p_k$  满足  $l \leq p_1 < p_2 < \dots < p_k \leq r$  且对每个  $i = 1, 2, \dots, k$  均有  $t_i = s_{p_i}$ 。

### Input

第一行包含两个正整数  $m$  ( $1 \leq m \leq 26$ ) 和  $n$  ( $1 \leq n \leq 200\,000$ ), 分别表示比特公司系统接受的命令的字符集大小和小 Q 事先准备的字符串的长度。

第二行包含一个长为  $n$  的只包含前  $m$  个小写英文字母的字符串, 表示小 Q 事先准备的字符串。

第三行包含一个正整数  $q$  ( $1 \leq q \leq 200\,000$ ), 表示攻防演练进行的轮数。

接下来  $q$  行, 每行包含两个正整数  $l$  和  $r$  ( $1 \leq l \leq r \leq n$ ), 表示本轮攻防演练中小 Q 配置字符串  $s_{l,r}$  为防火墙规则。

### Output

输出  $q$  行, 每行包含一个正整数, 表示本轮攻防演练中小 C 为了达成目标所需要输入的最短字符串的长度。

### Examples

stdin	stdout
2 6	2
abaabb	3
3	2
1 4	
2 5	
3 6	

## Problem C. 连锁商店

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

比特山是一个旅游胜地, 它一共有  $n$  个景点, 按照海拔高度从低到高依次编号为 1 到  $n$ 。为了更好地帮助游客们欣赏这里的风景, 人们在上面搭建了  $m$  条缆车路线。每条缆车路线只可能把游客们从某个海拔较低的景点运送到另一个海拔较高的景点。

在每个景点都有一家纪念品连锁商店, 其中第  $i$  个景点的商店隶属第  $c_i$  号公司, 两家连锁店  $(i, j)$  隶属同一公司当且仅当  $c_i = c_j$ 。每家公司都有新客优惠活动, 其中第  $i$  家公司对于新客的优惠红包为  $w_i$  元, 一旦领取了隶属该公司的某家连锁店的一份红包, 就不能再领取该公司所有分店的红包。

你正在 1 号景点, 你将会搭乘缆车去往各个景点, 每到一个景点, 你都可以领取该景点的连锁商店的新客优惠红包 (包括 1 号景点)。当然, 同一家公司的红包最多只能领一次。请写一个程序, 对于每个可能的终点  $k$ , 找到一条从 1 号景点出发到达  $k$  号景点的游览路线, 使得可以领取到总金额最多的优惠红包。

### Input

第一行包含两个正整数  $n, m$  ( $2 \leq n \leq 36, 1 \leq m \leq \frac{n(n-1)}{2}$ ), 分别表示景点的数量以及缆车路线的数量。

第二行包含  $n$  个正整数  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq n$ ), 依次表示每个景点的商店所隶属的公司。

第三行包含  $n$  个正整数  $w_1, w_2, \dots, w_n$  ( $1 \leq w_i \leq 10^6$ ), 依次表示每家公司的新客优惠红包的金额。

接下来  $m$  行, 每行两个正整数  $u_i, v_i$  ( $1 \leq u_i < v_i \leq n$ ), 表示一条缆车路线, 起点是景点  $u_i$ , 终点是景点  $v_i$ 。输入数据保证任意两个景点之间最多只有一条缆车路线, 且从 1 号景点出发可以到达任意一个景点。

### Output

输出  $n$  行, 第  $i$  行输出一个整数, 即从 1 号景点出发到达  $i$  号景点时, 领取的优惠红包的总金额的最大值。

## Examples

stdin	stdout
5 5	1
1 2 2 3 4	5
1 4 5 9 3	5
1 2	6
2 3	15
3 5	
1 4	
4 5	

## Problem D. 修建道路

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

比特镇有  $n$  个村庄, 编号依次为 1 到  $n$ 。现在需要修建  $n - 1$  条双向道路将这些村庄连通起来, 每条道路的端点只能是这  $n$  个村庄之一。准确地说, 如果要修建一条直接连接村庄  $i$  与村庄  $j$  的双向道路 ( $1 \leq i < j \leq n$ ), 那么需要支付  $\max_{i \leq k \leq j} \{a_k\}$  的费用。

请写一个程序, 找到一个总费用最小的修路方案, 使得任意两个村庄都能通过这些道路直接或间接到达。

### Input

第一行包含一个正整数  $n$  ( $1 \leq n \leq 200\,000$ ), 表示村庄的数量。

第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ )。

### Output

输出一行一个整数, 即所需的最小总费用。

### Examples

stdin	stdout
4 2 8 5 4	21

## Problem E. 被遗忘的计划

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

在比特超市有  $n$  种商品, 依次编号为 1 到  $n$ , 购买一件第  $i$  种商品的价格为  $i$  元, 其价值为  $v_i$ 。由于备货充足, 每种商品都可以购买任意非负整数件。

小 Q 计划购买恰好  $k$  ( $1 \leq k \leq 10^9$ ) 件商品, 并计算出了  $f_0, f_1, f_2, \dots, f_{n-1}$ , 其中  $f_i$  表示购买  $k$  件商品且商品价格之和对  $n$  取模的余数恰好为  $i$  的所有方案中商品价值之和的最大可能值。

一天过去之后, 健忘的小 Q 忘记了昨天做的购物计划中到底要买多少件物品, 请写一个程序根据小 Q 的回忆找到  $k$  的值。

### Input

第一行包含一个正整数  $n$  ( $1 \leq n \leq 1000$ ), 表示商品的种类数。

第二行包含  $n$  个整数  $v_1, v_2, \dots, v_n$  ( $1 \leq |v_i| \leq 10^9$ ), 依次表示每种商品的价值。

第三行包含  $n$  个整数  $f_0, f_1, \dots, f_{n-1}$  ( $-10^{18} \leq f_i \leq 10^{18}$ ), 表示小 Q 回忆出来的  $f$  的每一项的值。

### Output

输出一行一个整数  $k$  ( $1 \leq k \leq 10^9$ ), 即  $k$  的取值, 若有多个合法的  $k$ , 输出任意一个。如果小 Q 回忆有误, 即在  $[1, 10^9]$  内找不到对应的  $k$ , 请输出“-1”。

### Examples

stdin	stdout
3 1 2 3 6 4 5	2
3 1 2 3 -1 -2 -3	-1



## Problem F. 地图压缩

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

你正在参与一款 2D 游戏的地图绘制, 你拥有的美术素材是一张  $n \times n$  的像素图片, 从上到下依次编号为第 1 行到第  $n$  行, 从左往右依次编号为第 1 列到第  $n$  列, 其中第  $i$  行第  $j$  列的像素坐标为  $(i, j)$ , 它的内容可以用一个小写字母  $p_{i,j}$  来表示。

你希望从这张素材中裁剪出一个连续的矩形区域作为游戏的地图。你进行了  $q$  次尝试, 每次尝试中你将会选择以  $(x_1, y_1)$  为左上角、 $(x_2, y_2)$  为右下角的矩形部分 (包括端点) 作为游戏的地图。通过不断地尝试, 你发现可以通过应用“四方连续”的方法来压缩地图。“四方连续纹样”是指一个单位矩形纹样向上下左右四个方向反复连续循环排列所产生的纹样, 例如样例中以  $(1, 1)$  和  $(3, 7)$  为对角线端点的矩形可以看作以  $(1, 1)$  和  $(2, 3)$  为对角线端点的矩形向上下左右四个方向反复连续循环排列所产生的, 该矩形包含  $2 \times 3 = 6$  个像素点。注意, 纹样在生成地图的过程中可以超出边缘, 超出边缘的部分将被忽略。

请写一个程序, 对于每次尝试的地图, 找到包含像素点数量最少的单位矩形纹样, 使得它利用四方连续可以生成该次尝试的地图。

### Input

第一行包含两个正整数  $n$  和  $q$  ( $1 \leq n \leq 2000, 1 \leq q \leq 10000$ ), 分别表示素材的尺寸和尝试的次数。

接下来  $n$  行, 第  $i$  行包含一个长度为  $n$  的小写字符串, 其中第  $j$  个字符表示  $p_{i,j}$ 。

接下来  $q$  行, 每行四个正整数  $x_1, y_1, x_2, y_2$  ( $1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq n$ ), 依次表示每次尝试的矩形的对角线的端点坐标。

### Output

输出  $q$  行, 第  $i$  行输出一个正整数, 表示第  $i$  次尝试对应的最小单位纹样包含的像素点数量。

## Examples

stdin	stdout
8 5	6
abcabcab	14
defdefdb	5
abcabcab	2
aaaaaaaa	2
bbbbbbbb	
cccccccc	
bbbbbbbb	
cccccccc	
1 1 3 7	
1 2 3 8	
4 1 8 8	
5 2 7 6	
5 5 8 7	

## Problem G. 3G 网络

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

比特镇建镇多年一直没有通网, 工程师小 C 为了改善比特镇人民的生活, 立下了宏伟的目标, 致力于比特镇 3G 网络全域覆盖的实现。

比特镇可以被视为一个充分大的二维平面, 工程师小 C 敲定了  $n$  个建立 3G 网络基站的位置, 每个基站能够实现以基站为圆心的半径为  $r$  的圆内区域的 3G 网络覆盖。

现在工程师小 C 想知道, 当  $r$  足够大以至于比特镇的每一个角落都有 3G 网络覆盖时, 比特镇 3G 网络覆盖范围的面积与这  $n$  个基站的 3G 网络覆盖范围的面积之和的比值。

更形式化地描述这个问题, 记  $n$  个 3G 网络基站的位置分别为  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , 定义  $C_{i,r} = \{(x, y) \in \mathbb{R}^2 \mid (x - x_i)^2 + (y - y_i)^2 \leq r^2\}$  为第  $i$  个 3G 网络基站覆盖的范围, 你需要计算

$$f(r) = \frac{S(C_{1,r} \cup C_{2,r} \cup \dots \cup C_{n,r})}{S(C_{1,r}) + S(C_{2,r}) + \dots + S(C_{n,r})}$$

当  $r \rightarrow +\infty$  时  $f(r)$  的极限  $\lim_{r \rightarrow +\infty} f(r)$ , 其中  $S(X)$  表示平面点集  $X$  的面积。

### Input

第一行包含一个正整数  $n$  ( $1 \leq n \leq 2000$ ), 表示 3G 网络基站的个数。

接下来  $n$  行, 每行包含两个整数  $x, y$  ( $-10\,000 \leq x, y \leq 10\,000$ ), 表示 3G 网络基站建立的位置, 保证任意两个 3G 网络基站都不建在同一处。

### Output

输出一行, 包含一个实数表示  $\lim_{r \rightarrow +\infty} f(r)$ , 要求绝对误差不超过  $10^{-9}$ 。

也就是说, 如果你给出的答案是  $a$ , 标程给出的答案是  $b$ , 你的答案被认为是正确的当且仅当  $|a - b| \leq 10^{-9}$ 。

## Examples

stdin	stdout
1 0 0	1.0000000000000000
2 0 0 0 1	0.5000000000000000
3 0 0 0 1 1 0	0.3333333333333333
4 0 0 0 1 1 0 1 1	0.2500000000000000

## Problem H. 4G 网络

Input file:            stdin  
Output file:           stdout  
Time limit:           6 seconds  
Memory limit:        512 megabytes

经过多年的努力, 比特镇终于实现了 3G 网络的全域覆盖, 工程师小 C 想趁热打铁推行 4G 网络, 进一步提高比特镇人民的生活水平。

比特镇可以被视为一个充分大的二维平面, 工程师小 C 敲定了  $n$  个建立 4G 网络基站的位置, 每个基站能够实现以基站为圆心的半径为  $r$  的圆内区域的 4G 网络覆盖。

由于 4G 网络部署的成本远高于 3G 网络, 工程师小 C 准备对  $q$  个方案进行调研, 每个方案分别给出了 4G 网络基站覆盖范围的半径  $r$  的具体数值。

现在工程师小 C 想知道, 在每个方案中, 比特镇 4G 网络覆盖范围的面积与这  $n$  个基站的 4G 网络覆盖范围的面积之和的比值。

更形式化地描述这个问题, 记  $n$  个 4G 网络基站的位置分别为  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , 定义  $C_{i,r} = \{(x, y) \in \mathbb{R}^2 \mid (x - x_i)^2 + (y - y_i)^2 \leq r^2\}$  为第  $i$  个 4G 网络基站覆盖的范围, 你需要分别计算

$$f(r) = \frac{S(C_{1,r} \cup C_{2,r} \cup \dots \cup C_{n,r})}{S(C_{1,r}) + S(C_{2,r}) + \dots + S(C_{n,r})}$$

当  $r = r_1, r_2, \dots, r_q$  时  $f(r)$  的值  $f(r_1), f(r_2), \dots, f(r_q)$ , 其中  $S(X)$  表示平面点集  $X$  的面积。

### Input

第一行包含一个正整数  $n$  ( $1 \leq n \leq 2000$ ), 表示 4G 网络基站的个数。

接下来  $n$  行, 每行包含两个整数  $x, y$  ( $-10000 \leq x, y \leq 10000$ ), 表示 4G 网络基站建立的位置, 保证任意两个 4G 网络基站都不建在同一处。

下一行包含一个正整数  $q$  ( $1 \leq q \leq 2000$ ), 表示方案的个数。

接下来  $q$  行, 第  $i$  行包含一个整数  $r_i$  ( $1 \leq r_i \leq 20000$ ), 表示第  $i$  个方案中每个 4G 网络基站覆盖范围的半径。

### Output

输出  $q$  行, 第  $i$  行包含一个实数表示  $f(r_i)$ , 要求绝对误差不超过  $10^{-9}$ 。

也就是说, 如果你给出的答案是  $a$ , 标程给出的答案是  $b$ , 你的答案被认为是正确的当且仅当  $|a - b| \leq 10^{-9}$ 。

## Examples

stdin	stdout
1 0 0 1 1	1.0000000000000000
2 0 0 0 1 1 1	0.804498890522115
3 0 0 0 1 1 0 1 1	0.725769222409340
4 0 0 0 1 1 0 1 1 1 1	0.634076362068062

## Problem I. 驾驶卡丁车

Input file:            stdin  
Output file:           stdout  
Time limit:            1 second  
Memory limit:         512 megabytes

小 Q 正在设计一款 2D 卡丁车游戏，你的任务是帮助小 Q 实现其中的一部分功能。

在这款游戏中，游戏地图是一张  $n$  行  $m$  列的网格，从上到下依次编号为第 1 行到第  $n$  行，从左往右依次编号为第 1 列到第  $m$  列，其中第  $i$  行第  $j$  列的格子的坐标为  $(i, j)$ ，每个格子要么是可以通行的平地，要么是不可通行的障碍。

在地图上的某个平地格子处有一辆由玩家操控的卡丁车。卡丁车的移动速率为  $v$ ，并且一共有 8 种可能的朝向，分别为：

- “上”：前进一步时，将从  $(x, y)$  移动到  $(x - 1, y)$ 。
- “下”：前进一步时，将从  $(x, y)$  移动到  $(x + 1, y)$ 。
- “左”：前进一步时，将从  $(x, y)$  移动到  $(x, y - 1)$ 。
- “右”：前进一步时，将从  $(x, y)$  移动到  $(x, y + 1)$ 。
- “左上”：前进一步时，将从  $(x, y)$  移动到  $(x - 1, y - 1)$ 。
- “右上”：前进一步时，将从  $(x, y)$  移动到  $(x - 1, y + 1)$ 。
- “左下”：前进一步时，将从  $(x, y)$  移动到  $(x + 1, y - 1)$ 。
- “右下”：前进一步时，将从  $(x, y)$  移动到  $(x + 1, y + 1)$ 。

一开始卡丁车朝上位于某个平地格子处，其初始移动速率为  $v = 0$ 。接下来玩家将依次输入  $q$  条操作指令，每条操作指令是下列中的一种：

- “L”：卡丁车朝向往左转 45 度。
- “R”：卡丁车朝向往右转 45 度。
- “U”：卡丁车的速率由  $v$  增大至  $v + 1$ 。
- “D”：卡丁车的速率由  $v$  减小至  $\max(v - 1, 0)$ 。

在执行完每条操作指令后，卡丁车都会沿着其朝向前进  $v$  步，在移动结束后才会继续响应后续指令。在前进的过程中，如果某一步尝试驶入某个障碍格子或者尝试驶出地图，那么说明卡丁车发生了碰撞，它将就结束移动，在保持朝向的同时速率  $v$  降低为 0。特别要注意的是，当朝向是斜 45 度时，为了防止“穿模”现象的发生，如果卡丁车两侧都是障碍，那么卡丁车同样将被认为发生了碰撞。例如卡丁车朝向右下，现在将从  $(x, y)$  移动到  $(x + 1, y + 1)$ ，那么如果  $(x + 1, y)$  和  $(x, y + 1)$  都是障碍，则卡丁车发生了碰撞。

请写一个程序，在执行完每条操作指令后且卡丁车完成移动之后，汇报卡丁车的坐标以及这次移动过程中是否发生了碰撞。

## Input

第一行包含两个正整数  $n$  和  $m$  ( $1 \leq n, m \leq 50$ )，表示地图的尺寸。

接下来  $n$  行，第  $i$  行包含一个长度为  $m$  的字符串，其中第  $j$  个字符描述格子  $(i, j)$ 。如果它是“.”，则说明  $(i, j)$  是平地；如果它是“#”，则说明  $(i, j)$  是障碍；如果它是“\*”，则说明  $(i, j)$  是平地，且卡丁车位于此。输入数据保证存在恰好一个“\*”。

接下来一行包含一个正整数  $q$  ( $1 \leq q \leq 500$ )，表示指令的数量。

接下来一行包含一个长度为  $q$  的字符串，每个字符是四种指令中的一种，依次描述每条指令。

## Output

输出  $q$  行，第  $i$  行输出执行完第  $i$  条指令且卡丁车完成移动之后的相关信息。如果这一次卡丁车没有发生碰撞，那么输出“ $x y$ ”；如果这一次卡丁车发生了碰撞，那么输出“Crash!  $x y$ ”。其中  $x$  和  $y$  表示卡丁车的坐标为  $(x, y)$ 。



## Examples

stdin	stdout
5 4 .... ##.. ..#. .*.. .... 12 URULLULLLLUD	3 2 Crash! 3 2 Crash! 3 2 3 2 3 2 Crash! 3 2 3 2 3 2 3 2 4 3 4 3
10 10 ####...## ##.....# #...##... #...####.. #...####.. #..... #...####.. ...####.. *...####.. 16 UURUURRULUULLUUD	9 1 Crash! 9 1 9 1 8 2 6 4 Crash! 6 4 6 4 7 5 7 6 7 8 Crash! 7 10 7 10 7 10 6 10 4 10 3 10

## Problem J. 最大权边独立集

Input file:            stdin  
Output file:           stdout  
Time limit:            1.5 seconds  
Memory limit:         512 megabytes

给定一张  $n$  个点  $n - 1$  条边的连通无向图, 点的编号依次为 1 到  $n$ , 第  $i$  条边的边权为  $w_i$ 。

请往图中添加恰好  $k$  条边权为  $p$  的边, 使得最大权边独立集的边权和最大, 即在新图中找到一系列不含公共端点的边, 使得这些边的边权和最大。注意: 添加的边必须连接两个不同的点, 但是两个点之间允许存在多条边。

### Input

第一行包含三个整数  $n, k, p$  ( $2 \leq n \leq 100\,000, 0 \leq k \leq 100, 1 \leq p \leq 10^9$ ), 表示点数、要添加的边数以及要添加的边的权值。

接下来  $n - 1$  行, 第  $i$  行包含三个正整数  $u_i, v_i, w_i$  ( $1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^9$ ), 表示第  $i$  条边的端点是  $u_i$  和  $v_i$ , 其边权为  $w_i$ 。

输入数据保证图是连通图。

### Output

输出一行一个正整数, 即新图的最大权边独立集的边权和。

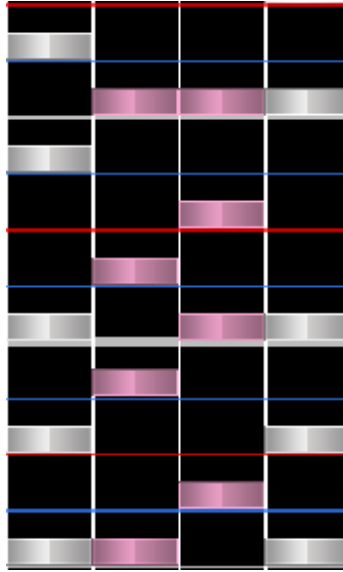
### Examples

stdin	stdout
4 1 7 1 2 8 2 3 4 2 4 6	15

## Problem K. 音乐游戏

Input file: `stdin`  
Output file: `stdout`  
Time limit: 1 second  
Memory limit: 512 megabytes

小 Q 最近一直在练习 `osu!mania` 的 4 键下落式模式, 现在给出一张 4 键下落式模式的谱面, 你需要帮小 Q 计算这张谱面中有多少个音符。



### Input

第一行包含一个正整数  $n$  ( $10 \leq n \leq 1000$ ), 表示给出的 4 键下落式模式谱面的长度。

接下来  $n$  行, 每行包含一个长为 6 的字符串, 保证第一个和最后一个字符是单个竖线 (“|”), 表示谱面展示区的两侧边界, 其余 4 个字符要么是单个空格 (“ ”), 表示对应位置没有音符, 要么是单个短横线 (“-”), 表示对应位置有一个音符。

### Output

输出一行, 包含一个整数, 表示谱面中的音符数量。

## Examples

stdin	stdout
10  -     ---   -     -     -    - --    -    - -    -    -- -	17

## Notes

对于样例，谱面如题图所示，沿着谱面的时间轴从下往上看，

- 第 1 行有 3 个音符；
- 第 2 行有 1 个音符；
- 第 3 行有 2 个音符；
- 第 4 行有 1 个音符；
- 第 5 行有 3 个音符；
- 第 6 行有 1 个音符；
- 第 7 行有 1 个音符；
- 第 8 行有 1 个音符；
- 第 9 行有 3 个音符；
- 第 10 行有 1 个音符；

加起来总共有 17 个音符。