

子鼠迎新赛

2020.1 by dst

一. 题目概况

中文题目名称	膜法石	宝石街	合并妹子
英文题目	mogic	gem	merge
可执行文件名	mogic	gem	merge
输入文件名	mogic.in	gem.in	merge.in
输出文件名	mogic.out	gem.out	merge.out
每个测试点时限	1.0 秒	3.0 秒	1.0 秒
测试点数目	20	20	20
每个测试点分值	5	5	5
附加样例文件	无	无	无
题目类型	传统	传统	传统
运行内存上限	256MiB	512MiB	256MiB

二. 编译选项

对于 C++ 语言	-lm
对于 C 语言	-lm
对于 Pascal 语言	

注意事项:

1. 本次比赛采用标准输入输出。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
3. 若题目没有特殊说明，则测试数据中所有在同一行的元素间用一个空格隔开，无多余空格，且在文件末尾有且只有一行换行。
4. 结果比较方式为全文比较（过滤行末空格及文末回车）。
5. `xor` 是按位异或运算符，在 C++/C 中用 `^` 表示，在 Pascal 中用 `xor` 表示。对于两个整数 a 和 b ， $a \text{ xor } b$ 表示 a 和 b 的异或和。
6. 对于实数 x ， $[x]$ 表示对 x 向上取整。
7. `mod` 是取余运算符，在 C++/C 中用 `%` 表示，在 Pascal 中用 `mod` 表示。对于两个整数 a 和 b ， $a \text{ mod } b$ 表示 a 除以 b 的余数。

膜法石 (mogic)

【问题描述】

dst 是一个喜欢研究算法的大魔法师。有一天，他在膜法森林里找到了 n 棵膜法树，其中第 i 棵膜法树可以生产无限块抗力为 a_i ，膜力为 b_i 的膜法石。同时，由于 dst 法力强大，他有 k 次机会，每次可以交换其中两棵膜法树能够产生的膜法石的膜力。

dst 可以吸收膜法石的能量。每吸收一块膜法石的能量，他的抗力就会减少对应膜法石的抗力，膜力就会增加对应膜法石的膜力。他的初始抗力为 m ，膜力为0。一旦他的抗力小于0，他就没法膜人了。现在，dst 想知道，自己在抗力不小于0的前提下，最大的膜力是多少。

注意，dst 必须先使用完所有的交换机会，再吸收膜法石的能量。

【输入格式】

输入共 3 行。

第 1 行包含 2 个正整数 n,m 和 1 个非负整数 k 。

第 2 行包含 n 个正整数，第 i 个数表示 a_i 。

第 3 行包含 n 个正整数，第 i 个数表示 b_i 。

【输出格式】

输出共 1 行，包含 1 个非负整数，表示 dst 最大的膜力。

【样例 1】

mogic.in	mogic.out
2 5 0 1 2 1 3	7

【样例 1 解释】

由于 $k = 0$ ，所以 dst 并没有交换机会。

吸收膜法树1的1块膜法石，吸收膜法树2的2块膜法石。

此时 dst 的抗力 $= 5 - 1 - 2 \times 2 = 0$ ，dst 的膜力 $= 1 + 2 \times 3 = 7$ 。

【样例 2】

mogic.in	mogic.out
2 5 1	15

1 2	
1 3	

【样例 2 解释】

dst 使用唯一的交换机会，交换膜法树1，膜法树2能够产生的膜法石的膜力。

吸收膜法树1的5块膜法石，吸收膜法树2的0块膜法石。

此时 dst 的抗力 = $5 - 1 \times 5 = 0$ ，dst 的膜力 = $3 \times 5 = 15$ 。

【数据范围】

数据点编号	$n, a_i, m \leq$	$k \leq$
1-4	10	0
5-10		4
11-12	100	0
13-16	10^3	
17-20		100

对于 100% 的数据， $1 < n \leq 10^3$; $1 \leq m, a_i \leq 10^3$; $1 \leq b_i \leq 10^9$; $0 \leq k \leq 100$ 。

宝石街 (gem)

【问题描述】

在 D 国 S 市 T 镇，坐落着一条长度为 n 的宝石街。把起点定为原点，以终点方向作为正方向建立数轴，那么对于每个 $i (1 \leq i \leq n)$ ，在点 i 处有 a_i 块宝石。身无分文的 dst 此时就站在原点，径直向终点走去，在行走的过程中经过每一个点，他都可以选择捡起任意块的宝石，或是扔掉任意块的宝石。设他在某一点拥有的宝石数为 k ，那么他走到下一个点需要的时间也为 k 。现在，dst 想知道，在行走时间恰好为 t （不需要到达终点）时，他最多可以拥有多少块宝石？

【输入格式】

由于本题部分测试点的数据范围较大，部分测试点的 a_i 将在程序内生成。

输入共 2 行。

第 1 行包含 3 个正整数 $n, t, type$ 。

若 $type = 1$ ，第 2 行包含 n 个正整数，第 i 个数表示 a_i 。

若 $type = 2$ ，第 2 行包含 2 个正整数 a_1 和 p 。对于每个 $i (2 \leq i \leq n)$ ，设 $x = a_{i-1} \text{ xor } (a_{i-1} \ll 13)$ ， $y = x \text{ xor } (x \gg 17)$ ，则 $a_i = (y \text{ xor } (y \ll 5)) \bmod p + 1$ 。

【输出格式】

输出共 1 行，包含 1 个非负整数，表示 dst 拥有的最多宝石数。

【样例 1】

gem.in	gem.out
5 12 1	12
2 5 2 5 2	

【样例 1 解释】

捡起点 2, 3, 4 上的所有宝石；不扔宝石。

【数据范围】

数据点编号	$n \leq$	$t \leq$	$type =$
1-2	7	10	1
3-6	10^3	100	
7-10		10^3	
11-13		10^{18}	
14-16	10^5		
17-20	5×10^7		2

对于 100% 的数据， $1 \leq n \leq 5 \times 10^7$ ； $1 \leq t \leq 10^{18}$ ； $1 \leq a_i, p \leq 10^3$ 。

合并妹子 (merge)

【问题描述】

jlb 有 n 个妹子。他现在要按一定顺序合并她们,使她们全部都合并在一起。合并的规则是,两堆妹子可以合并成一堆,那么显然需要合并 $n - 1$ 次。在合并过程中,合并在一起的一堆妹子叫做妹堆,对于每个妹堆 Q ,记妹子数为 t_Q ,并且有一个萌值 c_Q 。假设在一次合并中,妹堆 A 和妹堆 B 合并为妹堆 C ,且序号为 $(c_C \bmod n) + 1$ 的妹子合并前所在的妹堆为 D ,则 $c_A = \lceil \frac{c_C + t_D}{t_A} \rceil$, $c_B = \lceil \frac{c_C + t_D}{t_B} \rceil$ 。特殊地,开始时第 i 个妹子为独立的一个妹堆 i 。最终妹堆的萌值为 m ,jlb 想知道每个妹子最初的萌值 c_i 。

【输入格式】

输入共 n 行。
第 1 行包含两个正整数 n, m 。
接下来 $n - 1$ 行,第 i 行包括两个正整数 u_i 和 v_i ,表示第 i 次合并时,妹子 u_i 所在的妹堆和妹子 v_i 所在的妹堆合并。

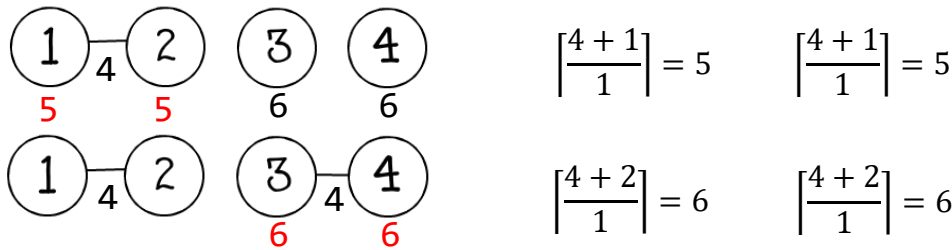
【输出格式】

输出共 1 行,包含 n 个整数,其中第 i 个整数表示 c_i 。

【样例 1】

merge.in	merge.out
4 5 1 2 3 4 1 4	5 5 6 6

【样例 1 解释】

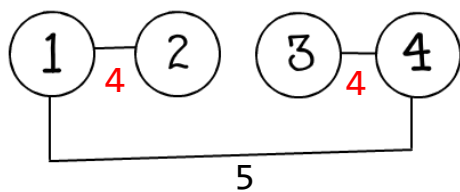


$\lceil \frac{4+1}{1} \rceil = 5$

$\lceil \frac{4+1}{1} \rceil = 5$

$\lceil \frac{4+2}{1} \rceil = 6$

$\lceil \frac{4+2}{1} \rceil = 6$



$$\left\lceil \frac{5+2}{2} \right\rceil = 4$$

$$\left\lceil \frac{5+2}{2} \right\rceil = 4$$

【数据范围】

对于 10% 的数据， $n = 2$ 。

对于 20% 的数据， $n \leq 5$ 。

对于 40% 的数据， $n \leq 500$ 。

对于 60% 的数据， $n \leq 5 \times 10^3$ 。

对于 100% 的数据， $1 \leq u_i, v_i \leq n \leq 10^5; 1 \leq m \leq 10^9$ 。