

NOI 休闲模拟

zzq

2018.06

题目名称	图	数列	走路
源程序文件名	g.c/cpp/pas	s.cpp	w.c/cpp/pas
输入文件名	g.in	N/A	w.in
输出文件名	g.out	N/A	w.out
时间限制	1s	0.3s	0.5s
是否捆绑测试	是	是	是
内存限制	1GB	1GB	1GB
是否有部分分	否	否	否
题目类型	传统	交互	传统
是否有附加文件	否	是	否
编译开关	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11

注意：

0. 今天代码长度限制为 20kb。
1. 下发文件夹下有若干样例。
2. 栈空间大小 = 256M。
3. 今天题目很简单，请大家不要假，认真做题。
4. AK 了不要 D 出题人，没 AK 也不要 D 出题人。
5. 按照剧本所有题目时限均为 std 运行最大点用时两倍以上。
6. 题目顺序真的是瞎放的，与难度无关。

Problem A. 图 (g.c/cpp/pas)

Input file: g.in
Output file: g.out
Time limit: 1 seconds
Memory limit: 1 gigabytes

老虎和蒜头是好朋友。

一天老虎在黑板上画了一个无向连通图，然后他跟蒜头说，我能把这个图的点用四种颜色染色，满足相邻点不同色。

蒜头不服气，在黑板上画了一个五个点的完全图。老虎跟蒜头说，这个图我能找到一个奇环，并且删掉这个奇环上的边之后图仍然联通。

蒜头发现他构不出反例了。蒜头很生气，他想让你也来解决一下这个问题。

一句话题意：对于一个无重边无自环的连通无向图，你可以选择两件事之一：把它的每个点用 1 2 3 4 这四种颜色染色，满足有边相连的点不同色；找到一个不重复顶点序列 $v_1, v_2 \dots v_k$ ($k \geq 3$, k 为奇数)，满足相邻两个点 (v_i 和 v_{i+1} , v_k 和 v_1) 之间有边，并且删掉这个环上的边之后图仍然联通。如果这两件事都干不了，请输出一行 wxh ak ioi2019 (这显然是事实)。

Input

第一行两个正整数 n, m 。

接下来 m 行两个整数 a, b ，表示 a 和 b 之间有一条无向边。

保证图连通，并且无重边无自环。

Output

如果你能把图四染色，请输出一行 $A \ c_1 \ c_2 \ \dots \ c_n$ ， c_i 表示第 i 个点的颜色 (为一个 $[1, 4]$ 的整数)。

如果你能找到一个删去其中的边后图仍然联通的奇环，请输出一行 $B \ k \ v_1 \ v_2 \ \dots \ v_k$ ， k 为环的大小， $v_1, v_2 \dots v_k$ 为环上的点 (详见题目描述)

如果这两件事都干不了，请输出一行 wxh ak ioi2019。

Examples

g.in	g.out
4 4 1 2 2 3 3 4 4 1	A 1 4 2 3
5 10 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5	B 3 1 3 5

合法的输出并不唯一。

大样例见选手文件夹下的 `ex_g3.in` 和 `ex_g3.out`。

Notes

对于所有数据， $1 \leq m \leq 300000$ 。

子任务编号	子任务分值	$n \leq$
1	20	5
2	20	17
3	20	60
4	40	300000

Problem B. 数列 (s.cpp)

Input file: `stdin`
 Output file: `stdout`
 Time limit: 0.3 seconds
 Memory limit: 1 gigabytes

这是一道交互题，本题仅支持 C++。

老虎和蒜头是好朋友。

虽然愚人节已经过了，为了让大家还能感受到被愚的乐趣，老虎又设置了一个神秘交互库来测试大家的智商。

交互库中有 n 个非负整数 $v_0, v_1 \dots v_{n-1}$ ，每个整数最多有 a 位（即在 $[0, 10^{a+1})$ ），你需要猜出这 n 个非负整数并返回。

你可以进行一种询问：给 n 个不超过 b 位（即在 $[0, 10^{b+1})$ ）的非负整数 $q_0, q_1 \dots q_{n-1}$ ，交互库会返回 $v_0q_0 + v_1q_1 + \dots + v_{n-1}q_{n-1}$ 。

你需要调用不超过 10 次询问，并返回猜出的序列。如果你成功通过了一个子任务中所有的数据，那么你会获得该子任务的分值，否则该子任务会获得 0 分。

由于数的位数比较大，交互时用 `std::string` 来传递高精度整数。你传递的和交互库返回的 `std::string` 需/会满足每个字符都在 0 到 9 之间，并且除了 0 用 0 表示外没有多余的前导零。

Interactor Notes

选手目录中有 `s.hpp`、`sample.cpp` 两个文件，以下是详细说明，如果懒得看的话你也可以直接参照 `sample.cpp` 编写代码。

你需要在你代码的开头 `#include "s.hpp"` 来与交互库交互，不应该实现 `main` 函数或进行任何文件输入输出，你只需要实现一个函数：`std::vector<std::string> guess(int n, int a, int b, int t)`，它接受 n 、 a 、 b 、 t ，其中 t 表示子任务编号，返回一个长度为 n 的 `std::vector` 表示你猜出的序列。

你可以使用交互库提供的一个函数：`std::string prod(std::vector<std::string> q)`，这个函数接受一个长度为 n 的由 $[0, 10^b)$ 的整数组成的 `std::vector` q ，返回 $v_0q_0 + v_1q_1 + \dots + v_{n-1}q_{n-1}$ 。

样例交互库（下发的 `s.hpp`）会从标准输入输入 n 、 a 、 b 、 t ，在第二行输入 n 个用空格分隔的非负整数 $v_0, v_1 \dots v_{n-1}$ 。编译只需将你的代码与 `s.hpp` 放在一个目录下正常编译即可。

Notes

子任务编号 (t)	子任务分值	$n =$	$a =$	$b =$
1	10	10	30	30
2	20	25	2	50
3	20	25	2	5
4	50	16	57	50

请注意数据范围均为等于，而不是小于等于。

交互库在正常交互时不会占用超过 0.05s 时间和 64MB 内存。

交互库是 non-adaptive 的，即交互库在调用你的 `guess` 函数之前就会生成好 n 个数，不会随着你的询问而改变。

请务必注意你传入的参数和返回值必须没有非数字字符和前导零，否则会直接获得 0 分。

Examples

见下发文件中的 `ex_s1.in` 到 `ex_s4.in`，分别对应四个子任务。

对于样例交互库而言正确的输出大概形如 `Well done! You made xxx queries.`

Problem C. 走路 (w.c/cpp/pas)

Input file: w.in
Output file: w.out
Time limit: 0.5 seconds
Memory limit: 1 gigabytes

老虎和蒜头是好朋友。

蒜头生活在数轴上，蒜头中学为 0，蒜头中学旁边有 n 家餐馆，编号为 i 的餐馆位置为 $i (i \in [1, n])$ 。

蒜头打算从蒜头中学所在的 0 号位置出发吃午餐，一开始他有 x 点行动力，他可以在数轴上左右移动。由于吃得越饱越难移动，如果他吃了 a 克食物，那么他每移动 1 单位距离（例如从位置 1 移动到位置 2）需要付出 $a + 1$ 点移动力。

如果某时刻蒜头在位置 $i \in [1, n]$ ，并且他没有进过第 i 家餐馆，那么他可以考虑进去吃饭。如果进去吃饭，蒜头会吃下 a_i 克食物（从而导致移动需要的移动力变大）。

蒜头打算吃得饱饱的，然后走回到蒜头中学（即回到 0）。蒜头想知道在能走回蒜头中学的前提下他最多能吃几克食物。根本不出校门也算可行的方案。

Input

输入第一行两个正整数 n, x ，表示餐馆的数量和初始行动力。

第二行 n 个非负整数 $a_1, a_2 \cdots a_n$ ，表示第 i 个餐馆进去之后会吃的食物克数。

Output

一行一个整数表示最多能吃几克食物。

Examples

w.in	w.out
5 10 1 1 1 1 1	2

文件夹下有大样例 ex_w2.in ex_w2.out。

Notes

对于所有数据， $0 \leq a_i \leq x$ 。

子任务编号	子任务分值	$n \leq$	$x \leq$
1	10	12	10^6
2	20	500	1000
3	25	5000	10^6
5	45	100000	10^6