# 6-1 单链表逆转（20 分）

本题要求实现一个函数，将给定的单链表逆转。

### 函数接口定义：

List Reverse( List L );

其中List结构定义如下：

typedef struct Node \*PtrToNode;

struct Node {

ElementType Data; /\* 存储结点数据 \*/

PtrToNode Next; /\* 指向下一个结点的指针 \*/

};

typedef PtrToNode List; /\* 定义单链表类型 \*/

L是给定单链表，函数Reverse要返回被逆转后的链表。

### 裁判测试程序样例：

#include <stdio.h>

#include <stdlib.h>

typedef int ElementType;

typedef struct Node \*PtrToNode;

struct Node {

ElementType Data;

PtrToNode Next;

};

typedef PtrToNode List;

List Read(); /\* 细节在此不表 \*/

void Print( List L ); /\* 细节在此不表 \*/

List Reverse( List L );

int main()

{

List L1, L2;

L1 = Read();

L2 = Reverse(L1);

Print(L1);

Print(L2);

return 0;

}

/\* 你的代码将被嵌在这里 \*/

### 输入样例：

5

1 3 4 5 2

### 输出样例：

1

2 5 4 3 1

# 7-1 最大子列和问题（20 分）

给定*K*个整数组成的序列{ *N*​1​​, *N*​2​​, ..., *N*​*K*​​ }，“连续子列”被定义为{ *N*​*i*​​, *N*​*i*+1​​, ..., *N*​*j*​​ }，其中 1≤*i*≤*j*≤*K*。“最大子列和”则被定义为所有连续子列元素的和中最大者。例如给定序列{ -2, 11, -4, 13, -5, -2 }，其连续子列{ 11, -4, 13 }有最大的和20。现要求你编写程序，计算给定整数序列的最大子列和。

本题旨在测试各种不同的算法在各种数据情况下的表现。各组测试数据特点如下：

* 数据1：与样例等价，测试基本正确性；
* 数据2：102个随机整数；
* 数据3：103个随机整数；
* 数据4：104个随机整数；
* 数据5：105个随机整数；

### 输入格式:

输入第1行给出正整数*K* (≤100000)；第2行给出*K*个整数，其间以空格分隔。

### 输出格式:

在一行中输出最大子列和。如果序列中所有整数皆为负数，则输出0。

### 输入样例:

6

-2 11 -4 13 -5 -2

### 输出样例:

20

# 7-2 一元多项式的乘法与加法运算（20 分）

设计函数分别求两个一元多项式的乘积与和。

### 输入格式:

输入分2行，每行分别先给出多项式非零项的个数，再以指数递降方式输入一个多项式非零项系数和指数（绝对值均为不超过1000的整数）。数字间以空格分隔。

### 输出格式:

输出分2行，分别以指数递降方式输出乘积多项式以及和多项式非零项的系数和指数。数字间以空格分隔，但结尾不能有多余空格。零多项式应输出0 0。

### 输入样例:

4 3 4 -5 2 6 1 -2 0

3 5 20 -7 4 3 1

### 输出样例:

15 24 -25 22 30 21 -10 20 -21 8 35 6 -33 5 14 4 -15 3 18 2 -6 1

5 20 -4 4 -5 2 9 1 -2 0

# 7-3 树的同构（25 分）

给定两棵树T1和T2。如果T1可以通过若干次左右孩子互换就变成T2，则我们称两棵树是“同构”的。例如图1给出的两棵树就是同构的，因为我们把其中一棵树的结点A、B、G的左右孩子互换后，就得到另外一棵树。而图2就不是同构的。

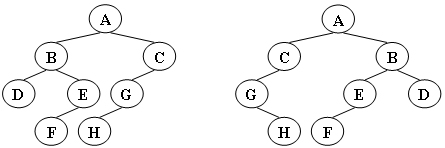


图1

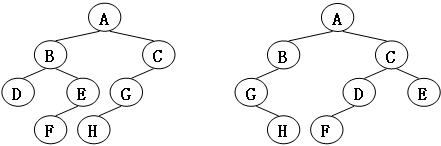


图2

现给定两棵树，请你判断它们是否是同构的。

### 输入格式:

输入给出2棵二叉树树的信息。对于每棵树，首先在一行中给出一个非负整数*N* (≤10)，即该树的结点数（此时假设结点从0到*N*−1编号）；随后*N*行，第*i*行对应编号第*i*个结点，给出该结点中存储的1个英文大写字母、其左孩子结点的编号、右孩子结点的编号。如果孩子结点为空，则在相应位置上给出“-”。给出的数据间用一个空格分隔。注意：题目保证每个结点中存储的字母是不同的。

### 输出格式:

如果两棵树是同构的，输出“Yes”，否则输出“No”。

### 输入样例1（对应图1）：

8

A 1 2

B 3 4

C 5 -

D - -

E 6 -

G 7 -

F - -

H - -

8

G - 4

B 7 6

F - -

A 5 1

H - -

C 0 -

D - -

E 2 -

### 输出样例1:

Yes

### 输入样例2（对应图2）：

8

B 5 7

F - -

A 0 3

C 6 -

H - -

D - -

G 4 -

E 1 -

8

D 6 -

B 5 -

E - -

H - -

C 0 2

G - 3

F - -

A 1 4

### 输出样例2:

No

# 7-4 是否同一棵二叉搜索树（25 分）

给定一个插入序列就可以唯一确定一棵二叉搜索树。然而，一棵给定的二叉搜索树却可以由多种不同的插入序列得到。例如分别按照序列{2, 1, 3}和{2, 3, 1}插入初始为空的二叉搜索树，都得到一样的结果。于是对于输入的各种插入序列，你需要判断它们是否能生成一样的二叉搜索树。

### 输入格式:

输入包含若干组测试数据。每组数据的第1行给出两个正整数*N* (≤10)和*L*，分别是每个序列插入元素的个数和需要检查的序列个数。第2行给出*N*个以空格分隔的正整数，作为初始插入序列。最后*L*行，每行给出*N*个插入的元素，属于*L*个需要检查的序列。

简单起见，我们保证每个插入序列都是1到*N*的一个排列。当读到*N*为0时，标志输入结束，这组数据不要处理。

### 输出格式:

对每一组需要检查的序列，如果其生成的二叉搜索树跟对应的初始序列生成的一样，输出“Yes”，否则输出“No”。

### 输入样例:

4 2

3 1 4 2

3 4 1 2

3 2 4 1

2 1

2 1

1 2

0

### 输出样例:

Yes

No

No

**鸣谢青岛大学周强老师补充测试数据！**

# 7-5 堆中的路径（25 分）

将一系列给定数字插入一个初始为空的小顶堆H[]。随后对任意给定的下标i，打印从H[i]到根结点的路径。

输入格式:

每组测试第1行包含2个正整数*N*和*M*(≤1000)，分别是插入元素的个数、以及需要打印的路径条数。下一行给出区间[-10000, 10000]内的*N*个要被插入一个初始为空的小顶堆的整数。最后一行给出*M*个下标。

输出格式:

对输入中给出的每个下标i，在一行中输出从H[i]到根结点的路径上的数据。数字间以1个空格分隔，行末不得有多余空格。

输入样例:

5 3

46 23 26 24 10

5 4 3

输出样例:

24 23 10

46 23 10

26 10

# 7-6 列出连通集（25 分）

给定一个有*N*个顶点和*E*条边的无向图，请用DFS和BFS分别列出其所有的连通集。假设顶点从0到*N*−1编号。进行搜索时，假设我们总是从编号最小的顶点出发，按编号递增的顺序访问邻接点。

### 输入格式:

输入第1行给出2个整数*N*(0<*N*≤10)和*E*，分别是图的顶点数和边数。随后*E*行，每行给出一条边的两个端点。每行中的数字之间用1空格分隔。

### 输出格式:

按照"{ *v*​1​​ *v*​2​​ ... *v*​*k*​​ }"的格式，每行输出一个连通集。先输出DFS的结果，再输出BFS的结果。

### 输入样例:

8 6

0 7

0 1

2 0

4 1

2 4

3 5

### 输出样例:

{ 0 1 4 2 7 }

{ 3 5 }

{ 6 }

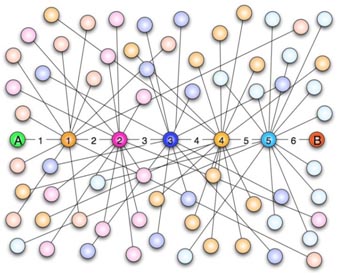
{ 0 1 2 7 4 }

{ 3 5 }

{ 6 }

# 7-7 六度空间（30 分）

“六度空间”理论又称作“六度分隔（Six Degrees of Separation）”理论。这个理论可以通俗地阐述为：“你和任何一个陌生人之间所间隔的人不会超过六个，也就是说，最多通过五个人你就能够认识任何一个陌生人。”如图1所示。

  
图1 六度空间示意图

“六度空间”理论虽然得到广泛的认同，并且正在得到越来越多的应用。但是数十年来，试图验证这个理论始终是许多社会学家努力追求的目标。然而由于历史的原因，这样的研究具有太大的局限性和困难。随着当代人的联络主要依赖于电话、短信、微信以及因特网上即时通信等工具，能够体现社交网络关系的一手数据已经逐渐使得“六度空间”理论的验证成为可能。

假如给你一个社交网络图，请你对每个节点计算符合“六度空间”理论的结点占结点总数的百分比。

### 输入格式:

输入第1行给出两个正整数，分别表示社交网络图的结点数*N*（1<*N*≤10​4​​，表示人数）、边数*M*（≤33×*N*，表示社交关系数）。随后的*M*行对应*M*条边，每行给出一对正整数，分别是该条边直接连通的两个结点的编号（节点从1到*N*编号）。

### 输出格式:

对每个结点输出与该结点距离不超过6的结点数占结点总数的百分比，精确到小数点后2位。每个结节点输出一行，格式为“结点编号:（空格）百分比%”。

### 输入样例:

10 9

1 2

2 3

3 4

4 5

5 6

6 7

7 8

8 9

9 10

### 输出样例:

1: 70.00%

2: 80.00%

3: 90.00%

4: 100.00%

5: 100.00%

6: 100.00%

7: 100.00%

8: 90.00%

9: 80.00%

10: 70.00%

# 7-8 哈利·波特的考试（25 分）

哈利·波特要考试了，他需要你的帮助。这门课学的是用魔咒将一种动物变成另一种动物的本事。例如将猫变成老鼠的魔咒是haha，将老鼠变成鱼的魔咒是hehe等等。反方向变化的魔咒就是简单地将原来的魔咒倒过来念，例如ahah可以将老鼠变成猫。另外，如果想把猫变成鱼，可以通过念一个直接魔咒lalala，也可以将猫变老鼠、老鼠变鱼的魔咒连起来念：hahahehe。

现在哈利·波特的手里有一本教材，里面列出了所有的变形魔咒和能变的动物。老师允许他自己带一只动物去考场，要考察他把这只动物变成任意一只指定动物的本事。于是他来问你：带什么动物去可以让最难变的那种动物（即该动物变为哈利·波特自己带去的动物所需要的魔咒最长）需要的魔咒最短？例如：如果只有猫、鼠、鱼，则显然哈利·波特应该带鼠去，因为鼠变成另外两种动物都只需要念4个字符；而如果带猫去，则至少需要念6个字符才能把猫变成鱼；同理，带鱼去也不是最好的选择。

### 输入格式:

输入说明：输入第1行给出两个正整数*N* (≤100)和*M*，其中*N*是考试涉及的动物总数，*M*是用于直接变形的魔咒条数。为简单起见，我们将动物按1~*N*编号。随后*M*行，每行给出了3个正整数，分别是两种动物的编号、以及它们之间变形需要的魔咒的长度(≤100)，数字之间用空格分隔。

### 输出格式:

输出哈利·波特应该带去考场的动物的编号、以及最长的变形魔咒的长度，中间以空格分隔。如果只带1只动物是不可能完成所有变形要求的，则输出0。如果有若干只动物都可以备选，则输出编号最小的那只。

### 输入样例:

6 11

3 4 70

1 2 1

5 4 50

2 6 50

5 6 60

1 3 70

4 6 60

3 6 80

5 1 100

2 4 60

5 2 80

### 输出样例:

4 70

# 7-9 旅游规划（25 分）

有了一张自驾旅游路线图，你会知道城市间的高速公路长度、以及该公路要收取的过路费。现在需要你写一个程序，帮助前来咨询的游客找一条出发地和目的地之间的最短路径。如果有若干条路径都是最短的，那么需要输出最便宜的一条路径。

### 输入格式:

输入说明：输入数据的第1行给出4个正整数*N*、*M*、*S*、*D*，其中*N*（2≤*N*≤500）是城市的个数，顺便假设城市的编号为0~(*N*−1)；*M*是高速公路的条数；*S*是出发地的城市编号；*D*是目的地的城市编号。随后的*M*行中，每行给出一条高速公路的信息，分别是：城市1、城市2、高速公路长度、收费额，中间用空格分开，数字均为整数且不超过500。输入保证解的存在。

### 输出格式:

在一行里输出路径的长度和收费总额，数字间以空格分隔，输出结尾不能有多余空格。

### 输入样例:

4 5 0 3

0 1 1 20

1 3 2 30

0 3 4 10

0 2 2 20

2 3 1 20

### 输出样例:

3 40

# 7-10 公路村村通（30 分）

现有村落间道路的统计数据表中，列出了有可能建设成标准公路的若干条道路的成本，求使每个村落都有公路连通所需要的最低成本。

### 输入格式:

输入数据包括城镇数目正整数*N*（≤1000）和候选道路数目*M*（≤3*N*）；随后的*M*行对应*M*条道路，每行给出3个正整数，分别是该条道路直接连通的两个城镇的编号以及该道路改建的预算成本。为简单起见，城镇从1到*N*编号。

### 输出格式:

输出村村通需要的最低成本。如果输入数据不足以保证畅通，则输出−1，表示需要建设更多公路。

### 输入样例:

6 15

1 2 5

1 3 3

1 4 7

1 5 4

1 6 2

2 3 4

2 4 6

2 5 2

2 6 6

3 4 6

3 5 1

3 6 1

4 5 10

4 6 8

5 6 3

### 输出样例:

12

# 7-11 关键活动（30 分）

假定一个工程项目由一组子任务构成，子任务之间有的可以并行执行，有的必须在完成了其它一些子任务后才能执行。“任务调度”包括一组子任务、以及每个子任务可以执行所依赖的子任务集。

比如完成一个专业的所有课程学习和毕业设计可以看成一个本科生要完成的一项工程，各门课程可以看成是子任务。有些课程可以同时开设，比如英语和C程序设计，它们没有必须先修哪门的约束；有些课程则不可以同时开设，因为它们有先后的依赖关系，比如C程序设计和数据结构两门课，必须先学习前者。

但是需要注意的是，对一组子任务，并不是任意的任务调度都是一个可行的方案。比如方案中存在“子任务A依赖于子任务B，子任务B依赖于子任务C，子任务C又依赖于子任务A”，那么这三个任务哪个都不能先执行，这就是一个不可行的方案。

任务调度问题中，如果还给出了完成每个子任务需要的时间，则我们可以算出完成整个工程需要的最短时间。在这些子任务中，有些任务即使推迟几天完成，也不会影响全局的工期；但是有些任务必须准时完成，否则整个项目的工期就要因此延误，这种任务就叫“关键活动”。

请编写程序判定一个给定的工程项目的任务调度是否可行；如果该调度方案可行，则计算完成整个工程项目需要的最短时间，并输出所有的关键活动。

### 输入格式:

输入第1行给出两个正整数*N*(≤100)和*M*，其中*N*是任务交接点（即衔接相互依赖的两个子任务的节点，例如：若任务2要在任务1完成后才开始，则两任务之间必有一个交接点）的数量。交接点按1~*N*编号，*M*是子任务的数量，依次编号为1~*M*。随后*M*行，每行给出了3个正整数，分别是该任务开始和完成涉及的交接点编号以及该任务所需的时间，整数间用空格分隔。

### 输出格式:

如果任务调度不可行，则输出0；否则第1行输出完成整个工程项目需要的时间，第2行开始输出所有关键活动，每个关键活动占一行，按格式“V->W”输出，其中V和W为该任务开始和完成涉及的交接点编号。关键活动输出的顺序规则是：任务开始的交接点编号小者优先，起点编号相同时，与输入时任务的顺序相反。

### 输入样例:

7 8

1 2 4

1 3 3

2 4 5

3 4 3

4 5 1

4 6 6

5 7 5

6 7 2

### 输出样例:

17

1->2

2->4

4->6

6->7

# 7-12 排序（25 分）

给定*N*个（长整型范围内的）整数，要求输出从小到大排序后的结果。

本题旨在测试各种不同的排序算法在各种数据情况下的表现。各组测试数据特点如下：

 数据1：只有1个元素；

 数据2：11个不相同的整数，测试基本正确性；

 数据3：103个随机整数；

 数据4：104个随机整数；

 数据5：105个随机整数；

 数据6：105个顺序整数；

 数据7：105个逆序整数；

 数据8：105个基本有序的整数；

 数据9：105个随机正整数，每个数字不超过1000。

### 输入格式:

输入第一行给出正整数*N*（≤10​5​​），随后一行给出*N*个（长整型范围内的）整数，其间以空格分隔。

### 输出格式:

在一行中输出从小到大排序后的结果，数字间以1个空格分隔，行末不得有多余空格。

### 输入样例:

11

4 981 10 -17 0 -20 29 50 8 43 -5

### 输出样例:

-20 -17 -5 0 4 8 10 29 43 50 981

# 7-13 统计工龄（20 分）

给定公司*N*名员工的工龄，要求按工龄增序输出每个工龄段有多少员工。

### 输入格式:

输入首先给出正整数*N*（≤10​5​​），即员工总人数；随后给出*N*个整数，即每个员工的工龄，范围在[0, 50]。

### 输出格式:

按工龄的递增顺序输出每个工龄的员工个数，格式为：“工龄:人数”。每项占一行。如果人数为0则不输出该项。

### 输入样例:

8

10 2 0 5 7 2 5 2

### 输出样例:

0:1

2:3

5:2

7:1

10:1

# 7-14 电话聊天狂人（25 分）

给定大量手机用户通话记录，找出其中通话次数最多的聊天狂人。

### 输入格式:

输入首先给出正整数*N*（≤10​5​​），为通话记录条数。随后*N*行，每行给出一条通话记录。简单起见，这里只列出拨出方和接收方的11位数字构成的手机号码，其中以空格分隔。

### 输出格式:

在一行中给出聊天狂人的手机号码及其通话次数，其间以空格分隔。如果这样的人不唯一，则输出狂人中最小的号码及其通话次数，并且附加给出并列狂人的人数。

### 输入样例:

4

13005711862 13588625832

13505711862 13088625832

13588625832 18087925832

15005713862 13588625832

### 输出样例:

13588625832 3

# 7-15 QQ帐户的申请与登陆（25 分）

实现QQ新帐户申请和老帐户登陆的简化版功能。最大挑战是：据说现在的QQ号码已经有10位数了。

### 输入格式:

输入首先给出一个正整数*N*（≤10​5​​），随后给出*N*行指令。每行指令的格式为：“命令符（空格）QQ号码（空格）密码”。其中命令符为“N”（代表New）时表示要新申请一个QQ号，后面是新帐户的号码和密码；命令符为“L”（代表Login）时表示是老帐户登陆，后面是登陆信息。QQ号码为一个不超过10位、但大于1000（据说QQ老总的号码是1001）的整数。密码为不小于6位、不超过16位、且不包含空格的字符串。

### 输出格式:

针对每条指令，给出相应的信息：

1）若新申请帐户成功，则输出“New: OK”；  
2）若新申请的号码已经存在，则输出“ERROR: Exist”；  
3）若老帐户登陆成功，则输出“Login: OK”；  
4）若老帐户QQ号码不存在，则输出“ERROR: Not Exist”；  
5）若老帐户密码错误，则输出“ERROR: Wrong PW”。

### 输入样例:

5

L 1234567890 myQQ@qq.com

N 1234567890 myQQ@qq.com

N 1234567890 myQQ@qq.com

L 1234567890 myQQ@qq

L 1234567890 myQQ@qq.com

### 输出样例:

ERROR: Not Exist

New: OK

ERROR: Exist

ERROR: Wrong PW

Login: OK

# 7-16 一元多项式求导（20 分）

设计函数求一元多项式的导数。

### 输入格式:

以指数递降方式输入多项式非零项系数和指数（绝对值均为不超过1000的整数）。数字间以空格分隔。

### 输出格式:

以与输入相同的格式输出导数多项式非零项的系数和指数。数字间以空格分隔，但结尾不能有多余空格。

### 输入样例:

3 4 -5 2 6 1 -2 0

### 输出样例:

12 3 -10 1 6 0

# 7-17 汉诺塔的非递归实现（25 分）

借助堆栈以非递归（循环）方式求解汉诺塔的问题（n, a, b, c），即将N个盘子从起始柱（标记为“a”）通过借助柱（标记为“b”）移动到目标柱（标记为“c”），并保证每个移动符合汉诺塔问题的要求。

### 输入格式:

输入为一个正整数N，即起始柱上的盘数。

### 输出格式:

每个操作（移动）占一行，按柱1 -> 柱2的格式输出。

### 输入样例:

3

### 输出样例:

a -> c

a -> b

c -> b

a -> c

b -> a

b -> c

a -> c

# 7-18 银行业务队列简单模拟（25 分）

设某银行有A、B两个业务窗口，且处理业务的速度不一样，其中A窗口处理速度是B窗口的2倍 —— 即当A窗口每处理完2个顾客时，B窗口处理完1个顾客。给定到达银行的顾客序列，请按业务完成的顺序输出顾客序列。假定不考虑顾客先后到达的时间间隔，并且当不同窗口同时处理完2个顾客时，A窗口顾客优先输出。

### 输入格式:

输入为一行正整数，其中第1个数字N(≤1000)为顾客总数，后面跟着N位顾客的编号。编号为奇数的顾客需要到A窗口办理业务，为偶数的顾客则去B窗口。数字间以空格分隔。

### 输出格式:

按业务处理完成的顺序输出顾客的编号。数字间以空格分隔，但最后一个编号后不能有多余的空格。

### 输入样例:

8 2 1 3 9 4 11 13 15

### 输出样例:

1 3 2 9 11 4 13 15

# 7-19 求链式线性表的倒数第K项（20 分）

给定一系列正整数，请设计一个尽可能高效的算法，查找倒数第K个位置上的数字。

### 输入格式:

输入首先给出一个正整数K，随后是若干正整数，最后以一个负整数表示结尾（该负数不算在序列内，不要处理）。

### 输出格式:

输出倒数第K个位置上的数据。如果这个位置不存在，输出错误信息NULL。

### 输入样例:

4 1 2 3 4 5 6 7 8 9 0 -1

### 输出样例:

7

# 7-20 表达式转换（25 分）

算术表达式有前缀表示法、中缀表示法和后缀表示法等形式。日常使用的算术表达式是采用中缀表示法，即二元运算符位于两个运算数中间。请设计程序将中缀表达式转换为后缀表达式。

### 输入格式:

输入在一行中给出不含空格的中缀表达式，可包含+、-、\*、\以及左右括号()，表达式不超过20个字符。

### 输出格式:

在一行中输出转换后的后缀表达式，要求不同对象（运算数、运算符号）之间以空格分隔，但结尾不得有多余空格。

### 输入样例:

2+3\*(7-4)+8/4

### 输出样例:

2 3 7 4 - \* + 8 4 / +

# 7-21 求前缀表达式的值（25 分）

算术表达式有前缀表示法、中缀表示法和后缀表示法等形式。前缀表达式指二元运算符位于两个运算数之前，例如2+3\*(7-4)+8/4的前缀表达式是：+ + 2 \* 3 - 7 4 / 8 4。请设计程序计算前缀表达式的结果值。

### 输入格式:

输入在一行内给出不超过30个字符的前缀表达式，只包含+、-、\*、\以及运算数，不同对象（运算数、运算符号）之间以空格分隔。

### 输出格式:

输出前缀表达式的运算结果，保留小数点后1位，或错误信息ERROR。

### 输入样例:

+ + 2 \* 3 - 7 4 / 8 4

### 输出样例:

13.0

# 7-22 堆栈模拟队列（25 分）

设已知有两个堆栈S1和S2，请用这两个堆栈模拟出一个队列Q。

所谓用堆栈模拟队列，实际上就是通过调用堆栈的下列操作函数:

* int IsFull(Stack S)：判断堆栈S是否已满，返回1或0；
* int IsEmpty (Stack S )：判断堆栈S是否为空，返回1或0；
* void Push(Stack S, ElementType item )：将元素item压入堆栈S；
* ElementType Pop(Stack S )：删除并返回S的栈顶元素。

实现队列的操作，即入队void AddQ(ElementType item)和出队ElementType DeleteQ()。

输入格式:

输入首先给出两个正整数N1和N2，表示堆栈S1和S2的最大容量。随后给出一系列的队列操作：A item表示将item入列（这里假设item为整型数字）；D表示出队操作；T表示输入结束。

输出格式:

对输入中的每个D操作，输出相应出队的数字，或者错误信息ERROR:Empty。如果入队操作无法执行，也需要输出ERROR:Full。每个输出占1行。

输入样例:

3 2

A 1 A 2 A 3 A 4 A 5 D A 6 D A 7 D A 8 D D D D T

输出样例:

ERROR:Full

1

ERROR:Full

2

3

4

7

8

ERROR:Empty

# 7-23 还原二叉树（25 分）

给定一棵二叉树的先序遍历序列和中序遍历序列，要求计算该二叉树的高度。

### 输入格式:

输入首先给出正整数N（≤50），为树中结点总数。下面两行先后给出先序和中序遍历序列，均是长度为N的不包含重复英文字母（区别大小写）的字符串。

### 输出格式:

输出为一个整数，即该二叉树的高度。

### 输入样例:

9

ABDFGHIEC

FDHGIBEAC

### 输出样例:

5

# 7-24 树种统计（25 分）

随着卫星成像技术的应用，自然资源研究机构可以识别每一棵树的种类。请编写程序帮助研究人员统计每种树的数量，计算每种树占总数的百分比。

### 输入格式:

输入首先给出正整数N（≤10​5​​），随后N行，每行给出卫星观测到的一棵树的种类名称。种类名称由不超过30个英文字母和空格组成（大小写不区分）。

### 输出格式:

按字典序递增输出各种树的种类名称及其所占总数的百分比，其间以空格分隔，保留小数点后4位。

### 输入样例:

29

Red Alder

Ash

Aspen

Basswood

Ash

Beech

Yellow Birch

Ash

Cherry

Cottonwood

Ash

Cypress

Red Elm

Gum

Hackberry

White Oak

Hickory

Pecan

Hard Maple

White Oak

Soft Maple

Red Oak

Red Oak

White Oak

Poplan

Sassafras

Sycamore

Black Walnut

Willow

### 输出样例:

Ash 13.7931%

Aspen 3.4483%

Basswood 3.4483%

Beech 3.4483%

Black Walnut 3.4483%

Cherry 3.4483%

Cottonwood 3.4483%

Cypress 3.4483%

Gum 3.4483%

Hackberry 3.4483%

Hard Maple 3.4483%

Hickory 3.4483%

Pecan 3.4483%

Poplan 3.4483%

Red Alder 3.4483%

Red Elm 3.4483%

Red Oak 6.8966%

Sassafras 3.4483%

Soft Maple 3.4483%

Sycamore 3.4483%

White Oak 10.3448%

Willow 3.4483%

Yellow Birch 3.4483%

# 7-25 朋友圈（25 分）

某学校有N个学生，形成M个俱乐部。每个俱乐部里的学生有着一定相似的兴趣爱好，形成一个朋友圈。一个学生可以同时属于若干个不同的俱乐部。根据“我的朋友的朋友也是我的朋友”这个推论可以得出，如果A和B是朋友，且B和C是朋友，则A和C也是朋友。请编写程序计算最大朋友圈中有多少人。

### 输入格式:

输入的第一行包含两个正整数N（≤30000）和M（≤1000），分别代表学校的学生总数和俱乐部的个数。后面的M行每行按以下格式给出1个俱乐部的信息，其中学生从1~N编号：

第i个俱乐部的人数Mi（空格）学生1（空格）学生2 … 学生Mi

### 输出格式:

输出给出一个整数，表示在最大朋友圈中有多少人。

### 输入样例:

7 4

3 1 2 3

2 1 4

3 5 6 7

1 6

### 输出样例:

4

# 7-26 Windows消息队列（25 分）

消息队列是Windows系统的基础。对于每个进程，系统维护一个消息队列。如果在进程中有特定事件发生，如点击鼠标、文字改变等，系统将把这个消息加到队列当中。同时，如果队列不是空的，这一进程循环地从队列中按照优先级获取消息。请注意优先级值低意味着优先级高。请编辑程序模拟消息队列，将消息加到队列中以及从队列中获取消息。

### 输入格式:

输入首先给出正整数N（≤10​5​​），随后N行，每行给出一个指令——GET或PUT，分别表示从队列中取出消息或将消息添加到队列中。如果指令是PUT，后面就有一个消息名称、以及一个正整数表示消息的优先级，此数越小表示优先级越高。消息名称是长度不超过10个字符且不含空格的字符串；题目保证队列中消息的优先级无重复，且输入至少有一个GET。

### 输出格式:

对于每个GET指令，在一行中输出消息队列中优先级最高的消息的名称和参数。如果消息队列中没有消息，输出EMPTY QUEUE!。对于PUT指令则没有输出。

### 输入样例:

9

PUT msg1 5

PUT msg2 4

GET

PUT msg3 2

PUT msg4 4

GET

GET

GET

GET

### 输出样例:

msg2

msg3

msg4

msg1

EMPTY QUEUE!

# 7-27 家谱处理（30 分）

人类学研究对于家族很感兴趣，于是研究人员搜集了一些家族的家谱进行研究。实验中，使用计算机处理家谱。为了实现这个目的，研究人员将家谱转换为文本文件。下面为家谱文本文件的实例：

John

Robert

Frank

Andrew

Nancy

David

家谱文本文件中，每一行包含一个人的名字。第一行中的名字是这个家族最早的祖先。家谱仅包含最早祖先的后代，而他们的丈夫或妻子不出现在家谱中。每个人的子女比父母多缩进2个空格。以上述家谱文本文件为例，John这个家族最早的祖先，他有两个子女Robert和Nancy，Robert有两个子女Frank和Andrew，Nancy只有一个子女David。

在实验中，研究人员还收集了家庭文件，并提取了家谱中有关两个人关系的陈述语句。下面为家谱中关系的陈述语句实例：

John is the parent of Robert

Robert is a sibling of Nancy

David is a descendant of Robert

研究人员需要判断每个陈述语句是真还是假，请编写程序帮助研究人员判断。

### 输入格式:

输入首先给出2个正整数*N*（2≤*N*≤100）和*M*（≤100），其中*N*为家谱中名字的数量，*M*为家谱中陈述语句的数量，输入的每行不超过70个字符。

名字的字符串由不超过10个英文字母组成。在家谱中的第一行给出的名字前没有缩进空格。家谱中的其他名字至少缩进2个空格，即他们是家谱中最早祖先（第一行给出的名字）的后代，且如果家谱中一个名字前缩进*k*个空格，则下一行中名字至多缩进*k*+2个空格。

在一个家谱中同样的名字不会出现两次，且家谱中没有出现的名字不会出现在陈述语句中。每句陈述语句格式如下，其中X和Y为家谱中的不同名字：

X is a child of Y

X is the parent of Y

X is a sibling of Y

X is a descendant of Y

X is an ancestor of Y

### 输出格式:

对于测试用例中的每句陈述语句，在一行中输出True，如果陈述为真，或False，如果陈述为假。

### 输入样例:

6 5

John

Robert

Frank

Andrew

Nancy

David

Robert is a child of John

Robert is an ancestor of Andrew

Robert is a sibling of Nancy

Nancy is the parent of Frank

John is a descendant of Andrew

### 输出样例:

True

True

True

False

False

# 7-28 搜索树判断（25 分）

对于二叉搜索树，我们规定任一结点的左子树仅包含严格小于该结点的键值，而其右子树包含大于或等于该结点的键值。如果我们交换每个节点的左子树和右子树，得到的树叫做镜像二叉搜索树。

现在我们给出一个整数键值序列，请编写程序判断该序列是否为某棵二叉搜索树或某镜像二叉搜索树的前序遍历序列，如果是，则输出对应二叉树的后序遍历序列。

### 输入格式:

输入的第一行包含一个正整数N（≤1000），第二行包含N个整数，为给出的整数键值序列，数字间以空格分隔。

### 输出格式:

输出的第一行首先给出判断结果，如果输入的序列是某棵二叉搜索树或某镜像二叉搜索树的前序遍历序列，则输出YES，否侧输出NO。如果判断结果是YES，下一行输出对应二叉树的后序遍历序列。数字间以空格分隔，但行尾不能有多余的空格。

### 输入样例1:

7

8 6 5 7 10 8 11

### 输出样例1:

YES

5 7 6 8 11 10 8

### 输入样例2:

7

8 6 8 5 10 9 11

### 输出样例2:

NO

# 7-29 修理牧场（25 分）

农夫要修理牧场的一段栅栏，他测量了栅栏，发现需要*N*块木头，每块木头长度为整数*L*​*i*​​个长度单位，于是他购买了一条很长的、能锯成*N*块的木头，即该木头的长度是*L*​*i*​​的总和。

但是农夫自己没有锯子，请人锯木的酬金跟这段木头的长度成正比。为简单起见，不妨就设酬金等于所锯木头的长度。例如，要将长度为20的木头锯成长度为8、7和5的三段，第一次锯木头花费20，将木头锯成12和8；第二次锯木头花费12，将长度为12的木头锯成7和5，总花费为32。如果第一次将木头锯成15和5，则第二次锯木头花费15，总花费为35（大于32）。

请编写程序帮助农夫计算将木头锯成*N*块的最少花费。

### 输入格式:

输入首先给出正整数*N*（≤10​4​​），表示要将木头锯成*N*块。第二行给出*N*个正整数（≤50），表示每段木块的长度。

### 输出格式:

输出一个整数，即将木头锯成*N*块的最少花费。

### 输入样例:

8

4 5 1 2 1 3 1 1

### 输出样例:

49

# 7-30 目录树（30 分）

在ZIP归档文件中，保留着所有压缩文件和目录的相对路径和名称。当使用WinZIP等GUI软件打开ZIP归档文件时，可以从这些信息中重建目录的树状结构。请编写程序实现目录的树状结构的重建工作。

输入格式:

输入首先给出正整数N（≤10​4​​），表示ZIP归档文件中的文件和目录的数量。随后N行，每行有如下格式的文件或目录的相对路径和名称（每行不超过260个字符）：

* 路径和名称中的字符仅包括英文字母（区分大小写）；
* 符号“\”仅作为路径分隔符出现；
* 目录以符号“\”结束；
* 不存在重复的输入项目；
* 整个输入大小不超过2MB。

输出格式:

假设所有的路径都相对于root目录。从root目录开始，在输出时每个目录首先输出自己的名字，然后以字典序输出所有子目录，然后以字典序输出所有文件。注意，在输出时，应根据目录的相对关系使用空格进行缩进，每级目录或文件比上一级多缩进2个空格。

输入样例:

7

b

c\

ab\cd

a\bc

ab\d

a\d\a

a\d\z\

输出样例:

root

a

d

z

a

bc

ab

cd

d

c

b

# 7-31 笛卡尔树（25 分）

笛卡尔树是一种特殊的二叉树，其结点包含两个关键字K1和K2。首先笛卡尔树是关于K1的二叉搜索树，即结点左子树的所有K1值都比该结点的K1值小，右子树则大。其次所有结点的K2关键字满足优先队列（不妨设为最小堆）的顺序要求，即该结点的K2值比其子树中所有结点的K2值小。给定一棵二叉树，请判断该树是否笛卡尔树。

### 输入格式:

输入首先给出正整数N（≤1000），为树中结点的个数。随后N行，每行给出一个结点的信息，包括：结点的K1值、K2值、左孩子结点编号、右孩子结点编号。设结点从0~(N-1)顺序编号。若某结点不存在孩子结点，则该位置给出−1。

### 输出格式:

输出YES如果该树是一棵笛卡尔树；否则输出NO。

### 输入样例1:

6

8 27 5 1

9 40 -1 -1

10 20 0 3

12 21 -1 4

15 22 -1 -1

5 35 -1 -1

### 输出样例1:

YES

### 输入样例2:

6

8 27 5 1

9 40 -1 -1

10 20 0 3

12 11 -1 4

15 22 -1 -1

50 35 -1 -1

### 输出样例2:

NO

作者: DS课程组

单位: 浙江大学

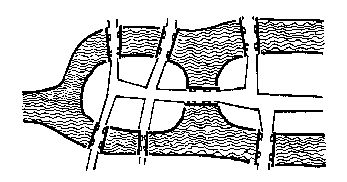
时间限制: 400ms

内存限制: 64MB

代码长度限制: 16KB

# 7-32 哥尼斯堡的“七桥问题”（25 分）

哥尼斯堡是位于普累格河上的一座城市，它包含两个岛屿及连接它们的七座桥，如下图所示。



可否走过这样的七座桥，而且每桥只走过一次？瑞士数学家欧拉(Leonhard Euler，1707—1783)最终解决了这个问题，并由此创立了拓扑学。

这个问题如今可以描述为判断欧拉回路是否存在的问题。欧拉回路是指不令笔离开纸面，可画过图中每条边仅一次，且可以回到起点的一条回路。现给定一个无向图，问是否存在欧拉回路？

### 输入格式:

输入第一行给出两个正整数，分别是节点数*N* (1≤*N*≤1000)和边数*M*；随后的*M*行对应*M*条边，每行给出一对正整数，分别是该条边直接连通的两个节点的编号（节点从1到*N*编号）。

### 输出格式:

若欧拉回路存在则输出1，否则输出0。

### 输入样例1:

6 10

1 2

2 3

3 1

4 5

5 6

6 4

1 4

1 6

3 4

3 6

### 输出样例1:

1

### 输入样例2:

5 8

1 2

1 3

2 3

2 4

2 5

5 3

5 4

3 4

### 输出样例2:

0

作者: DS课程组

单位: 浙江大学

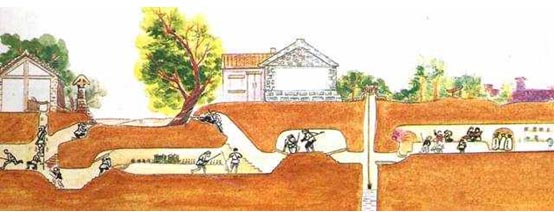
时间限制: 400ms

内存限制: 64MB

代码长度限制: 16KB

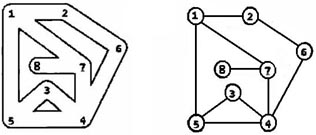
# 7-33 地下迷宫探索（30 分）

地道战是在抗日战争时期，在华北平原上抗日军民利用地道打击日本侵略者的作战方式。地道网是房连房、街连街、村连村的地下工事，如下图所示。



我们在回顾前辈们艰苦卓绝的战争生活的同时，真心钦佩他们的聪明才智。在现在和平发展的年代，对多数人来说，探索地下通道或许只是一种娱乐或者益智的游戏。本实验案例以探索地下通道迷宫作为内容。

假设有一个地下通道迷宫，它的通道都是直的，而通道所有交叉点（包括通道的端点）上都有一盏灯和一个开关。请问你如何从某个起点开始在迷宫中点亮所有的灯并回到起点？



### 输入格式:

输入第一行给出三个正整数，分别表示地下迷宫的节点数*N*（1<*N*≤1000，表示通道所有交叉点和端点）、边数*M*（≤3000，表示通道数）和探索起始节点编号*S*（节点从1到*N*编号）。随后的*M*行对应*M*条边（通道），每行给出一对正整数，分别是该条边直接连通的两个节点的编号。

### 输出格式:

若可以点亮所有节点的灯，则输出从*S*开始并以*S*结束的包含所有节点的序列，序列中相邻的节点一定有边（通道）；否则虽然不能点亮所有节点的灯，但还是输出点亮部分灯的节点序列，最后输出0，此时表示迷宫不是连通图。

由于深度优先遍历的节点序列是不唯一的，为了使得输出具有唯一的结果，我们约定以节点小编号优先的次序访问（点灯）。在点亮所有可以点亮的灯后，以原路返回的方式回到起点。

### 输入样例1:

6 8 1

1 2

2 3

3 4

4 5

5 6

6 4

3 6

1 5

### 输出样例1:

1 2 3 4 5 6 5 4 3 2 1

### 输入样例2:

6 6 6

1 2

1 3

2 3

5 4

6 5

6 4

### 输出样例2:

6 4 5 4 6 0

# 7-34 任务调度的合理性（25 分）

假定一个工程项目由一组子任务构成，子任务之间有的可以并行执行，有的必须在完成了其它一些子任务后才能执行。“任务调度”包括一组子任务、以及每个子任务可以执行所依赖的子任务集。

比如完成一个专业的所有课程学习和毕业设计可以看成一个本科生要完成的一项工程，各门课程可以看成是子任务。有些课程可以同时开设，比如英语和C程序设计，它们没有必须先修哪门的约束；有些课程则不可以同时开设，因为它们有先后的依赖关系，比如C程序设计和数据结构两门课，必须先学习前者。

但是需要注意的是，对一组子任务，并不是任意的任务调度都是一个可行的方案。比如方案中存在“子任务A依赖于子任务B，子任务B依赖于子任务C，子任务C又依赖于子任务A”，那么这三个任务哪个都不能先执行，这就是一个不可行的方案。你现在的工作是写程序判定任何一个给定的任务调度是否可行。

### 输入格式:

输入说明：输入第一行给出子任务数*N*（≤100），子任务按1~*N*编号。随后*N*行，每行给出一个子任务的依赖集合：首先给出依赖集合中的子任务数*K*，随后给出*K*个子任务编号，整数之间都用空格分隔。

### 输出格式:

如果方案可行，则输出1，否则输出0。

### 输入样例1:

12

0

0

2 1 2

0

1 4

1 5

2 3 6

1 3

2 7 8

1 7

1 10

1 7

### 输出样例1:

1

### 输入样例2:

5

1 4

2 1 4

2 2 5

1 3

0

### 输出样例2:

0

# 7-35 城市间紧急救援（25 分）

作为一个城市的应急救援队伍的负责人，你有一张特殊的全国地图。在地图上显示有多个分散的城市和一些连接城市的快速道路。每个城市的救援队数量和每一条连接两个城市的快速道路长度都标在地图上。当其他城市有紧急求助电话给你的时候，你的任务是带领你的救援队尽快赶往事发地，同时，一路上召集尽可能多的救援队。

### 输入格式:

输入第一行给出4个正整数*N*、*M*、*S*、*D*，其中*N*（2≤*N*≤500）是城市的个数，顺便假设城市的编号为0 ~ (*N*−1)；*M*是快速道路的条数；*S*是出发地的城市编号；*D*是目的地的城市编号。

第二行给出*N*个正整数，其中第*i*个数是第*i*个城市的救援队的数目，数字间以空格分隔。随后的*M*行中，每行给出一条快速道路的信息，分别是：城市1、城市2、快速道路的长度，中间用空格分开，数字均为整数且不超过500。输入保证救援可行且最优解唯一。

### 输出格式:

第一行输出最短路径的条数和能够召集的最多的救援队数量。第二行输出从*S*到*D*的路径中经过的城市编号。数字间以空格分隔，输出结尾不能有多余空格。

### 输入样例:

4 5 0 3

20 30 40 10

0 1 1

1 3 2

0 3 3

0 2 2

2 3 2

### 输出样例:

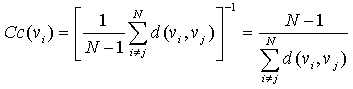
2 60

0 1 3

# 7-36 社交网络图中结点的“重要性”计算（30 分）

在社交网络中，个人或单位（结点）之间通过某些关系（边）联系起来。他们受到这些关系的影响，这种影响可以理解为网络中相互连接的结点之间蔓延的一种相互作用，可以增强也可以减弱。而结点根据其所处的位置不同，其在网络中体现的重要性也不尽相同。

“紧密度中心性”是用来衡量一个结点到达其它结点的“快慢”的指标，即一个有较高中心性的结点比有较低中心性的结点能够更快地（平均意义下）到达网络中的其它结点，因而在该网络的传播过程中有更重要的价值。在有*N*个结点的网络中，结点*v*​*i*​​的“紧密度中心性”*Cc*(*v*​*i*​​)数学上定义为*v*​*i*​​到其余所有结点*v*​*j*​​ (*j*≠*i*) 的最短距离*d*(*v*​*i*​​,*v*​*j*​​)的平均值的倒数：



对于非连通图，所有结点的紧密度中心性都是0。

给定一个无权的无向图以及其中的一组结点，计算这组结点中每个结点的紧密度中心性。

### 输入格式:

输入第一行给出两个正整数*N*和*M*，其中*N*（≤10​4​​）是图中结点个数，顺便假设结点从1到*N*编号；*M*（≤10​5​​）是边的条数。随后的*M*行中，每行给出一条边的信息，即该边连接的两个结点编号，中间用空格分隔。最后一行给出需要计算紧密度中心性的这组结点的个数*K*（≤100）以及*K*个结点编号，用空格分隔。

### 输出格式:

按照Cc(i)=x.xx的格式输出*K*个给定结点的紧密度中心性，每个输出占一行，结果保留到小数点后2位。

### 输入样例:

9 14

1 2

1 3

1 4

2 3

3 4

4 5

4 6

5 6

5 7

5 8

6 7

6 8

7 8

7 9

3 3 4 9

### 输出样例:

Cc(3)=0.47

Cc(4)=0.62

Cc(9)=0.35

# 7-37 模拟EXCEL排序（25 分）

Excel可以对一组纪录按任意指定列排序。现请编写程序实现类似功能。

### 输入格式:

输入的第一行包含两个正整数*N*(≤10​5​​) 和*C*，其中*N*是纪录的条数，*C*是指定排序的列号。之后有 *N*行，每行包含一条学生纪录。每条学生纪录由学号（6位数字，保证没有重复的学号）、姓名（不超过8位且不包含空格的字符串）、成绩（[0, 100]内的整数）组成，相邻属性用1个空格隔开。

### 输出格式:

在*N*行中输出按要求排序后的结果，即：当*C*=1时，按学号递增排序；当*C*=2时，按姓名的非递减字典序排序；当*C*=3时，按成绩的非递减排序。当若干学生具有相同姓名或者相同成绩时，则按他们的学号递增排序。

### 输入样例:

3 1

000007 James 85

000010 Amy 90

000001 Zoe 60

### 输出样例:

000001 Zoe 60

000007 James 85

000010 Amy 90

# 7-38 寻找大富翁（25 分）

2015年胡润研究院的调查显示，截至2014年9月，个人资产在600万元以上高净值人群达290万人。假设给出N个人的个人资产值，请快速找出资产排前M位的大富翁。

### 输入格式:

输入首先给出两个正整数*N*（≤10​6​​）和*M*（≤10），其中*N*为总人数，*M*为需要找出的大富翁数；接下来一行给出*N*个人的个人资产值，以百万元为单位，为不超过长整型范围的整数。数字间以空格分隔。

### 输出格式:

在一行内按非递增顺序输出资产排前*M*位的大富翁的个人资产值。数字间以空格分隔，但结尾不得有多余空格。

### 输入样例:

8 3

8 12 7 3 20 9 5 18

### 输出样例:

20 18 12

# 7-39 魔法优惠券（25 分）

在火星上有个魔法商店，提供魔法优惠券。每个优惠劵上印有一个整数面值K，表示若你在购买某商品时使用这张优惠劵，可以得到K倍该商品价值的回报！该商店还免费赠送一些有价值的商品，但是如果你在领取免费赠品的时候使用面值为正的优惠劵，则必须倒贴给商店K倍该商品价值的金额…… 但是不要紧，还有面值为负的优惠劵可以用！（真是神奇的火星）

例如，给定一组优惠劵，面值分别为1、2、4、-1；对应一组商品，价值为火星币M$7、6、-2、-3，其中负的价值表示该商品是免费赠品。我们可以将优惠劵3用在商品1上，得到M$28的回报；优惠劵2用在商品2上，得到M$12的回报；优惠劵4用在商品4上，得到M$3的回报。但是如果一不小心把优惠劵3用在商品4上，你必须倒贴给商店M$12。同样，当你一不小心把优惠劵4用在商品1上，你必须倒贴给商店M$7。

规定每张优惠券和每件商品都只能最多被使用一次，求你可以得到的最大回报。

### 输入格式:

输入有两行。第一行首先给出优惠劵的个数N，随后给出N个优惠劵的整数面值。第二行首先给出商品的个数M，随后给出M个商品的整数价值。N和M在[1, 10​6​​]之间，所有的数据大小不超过2​30​​，数字间以空格分隔。

### 输出格式:

输出可以得到的最大回报。

### 输入样例:

4 1 2 4 -1

4 7 6 -2 -3

### 输出样例:

43

# 7-40 奥运排行榜（25 分）

每年奥运会各大媒体都会公布一个排行榜，但是细心的读者发现，不同国家的排行榜略有不同。比如中国金牌总数列第一的时候，中国媒体就公布“金牌榜”；而美国的奖牌总数第一，于是美国媒体就公布“奖牌榜”。如果人口少的国家公布一个“国民人均奖牌榜”，说不定非洲的国家会成为榜魁…… 现在就请你写一个程序，对每个前来咨询的国家按照对其最有利的方式计算它的排名。

### 输入格式:

输入的第一行给出两个正整数*N*和*M*（≤224，因为世界上共有224个国家和地区），分别是参与排名的国家和地区的总个数、以及前来咨询的国家的个数。为简单起见，我们把国家从0 ~ *N*−1编号。之后有*N*行输入，第*i*行给出编号为*i*−1的国家的金牌数、奖牌数、国民人口数（单位为百万），数字均为[0,1000]区间内的整数，用空格分隔。最后面一行给出*M*个前来咨询的国家的编号，用空格分隔。

### 输出格式:

在一行里顺序输出前来咨询的国家的排名:计算方式编号。其排名按照对该国家最有利的方式计算；计算方式编号为：金牌榜=1，奖牌榜=2，国民人均金牌榜=3，国民人均奖牌榜=4。输出间以空格分隔，输出结尾不能有多余空格。

若某国在不同排名方式下有相同名次，则输出编号最小的计算方式。

### 输入样例:

4 4

51 100 1000

36 110 300

6 14 32

5 18 40

0 1 2 3

### 输出样例:

1:1 1:2 1:3 1:4

# 7-41 PAT排名汇总（25 分）

计算机程序设计能力考试（Programming Ability Test，简称PAT）旨在通过统一组织的在线考试及自动评测方法客观地评判考生的算法设计与程序设计实现能力，科学的评价计算机程序设计人才，为企业选拔人才提供参考标准（网址http://www.patest.cn）。

每次考试会在若干个不同的考点同时举行，每个考点用局域网，产生本考点的成绩。考试结束后，各个考点的成绩将即刻汇总成一张总的排名表。

现在就请你写一个程序自动归并各个考点的成绩并生成总排名表。

### 输入格式:

输入的第一行给出一个正整数N（≤100），代表考点总数。随后给出N个考点的成绩，格式为：首先一行给出正整数K（≤300），代表该考点的考生总数；随后K行，每行给出1个考生的信息，包括考号（由13位整数字组成）和得分（为[0,100]区间内的整数），中间用空格分隔。

### 输出格式:

首先在第一行里输出考生总数。随后输出汇总的排名表，每个考生的信息占一行，顺序为：考号、最终排名、考点编号、在该考点的排名。其中考点按输入给出的顺序从1到N编号。考生的输出须按最终排名的非递减顺序输出，获得相同分数的考生应有相同名次，并按考号的递增顺序输出。

### 输入样例:

2

5

1234567890001 95

1234567890005 100

1234567890003 95

1234567890002 77

1234567890004 85

4

1234567890013 65

1234567890011 25

1234567890014 100

1234567890012 85

### 输出样例:

9

1234567890005 1 1 1

1234567890014 1 2 1

1234567890001 3 1 2

1234567890003 3 1 2

1234567890004 5 1 4

1234567890012 5 2 2

1234567890002 7 1 5

1234567890013 8 2 3

1234567890011 9 2 4

# 7-42 整型关键字的散列映射（25 分）

给定一系列整型关键字和素数*P*，用除留余数法定义的散列函数将关键字映射到长度为*P*的散列表中。用线性探测法解决冲突。

### 输入格式:

输入第一行首先给出两个正整数*N*（≤1000）和*P*（≥*N*的最小素数），分别为待插入的关键字总数、以及散列表的长度。第二行给出*N*个整型关键字。数字间以空格分隔。

### 输出格式:

在一行内输出每个整型关键字在散列表中的位置。数字间以空格分隔，但行末尾不得有多余空格。

### 输入样例:

4 5

24 15 61 88

### 输出样例:

4 0 1 3

# 7-43 字符串关键字的散列映射（25 分）

给定一系列由大写英文字母组成的字符串关键字和素数*P*，用移位法定义的散列函数*H*(*Key*)将关键字*Key*中的最后3个字符映射为整数，每个字符占5位；再用除留余数法将整数映射到长度为*P*的散列表中。例如将字符串AZDEG插入长度为1009的散列表中，我们首先将26个大写英文字母顺序映射到整数0~25；再通过移位将其映射为3×32​2​​+4×32+6=3206；然后根据表长得到，即是该字符串的散列映射位置。

发生冲突时请用平方探测法解决。

### 输入格式:

输入第一行首先给出两个正整数*N*（≤500）和*P*（≥2*N*的最小素数），分别为待插入的关键字总数、以及散列表的长度。第二行给出*N*个字符串关键字，每个长度不超过8位，其间以空格分隔。

### 输出格式:

在一行内输出每个字符串关键字在散列表中的位置。数字间以空格分隔，但行末尾不得有多余空格。

### 输入样例1:

4 11

HELLO ANNK ZOE LOLI

### 输出样例1:

3 10 4 0

### 输入样例2:

6 11

LLO ANNA NNK ZOJ INNK AAA

### 输出样例2:

3 0 10 9 6 1

# 7-44 基于词频的文件相似度（30 分）

实现一种简单原始的文件相似度计算，即以两文件的公共词汇占总词汇的比例来定义相似度。为简化问题，这里不考虑中文（因为分词太难了），只考虑长度不小于3、且不超过10的英文单词，长度超过10的只考虑前10个字母。

### 输入格式:

输入首先给出正整数*N*（≤100），为文件总数。随后按以下格式给出每个文件的内容：首先给出文件正文，最后在一行中只给出一个字符#，表示文件结束。在*N*个文件内容结束之后，给出查询总数*M*（≤10​4​​），随后*M*行，每行给出一对文件编号，其间以空格分隔。这里假设文件按给出的顺序从1到*N*编号。

### 输出格式:

针对每一条查询，在一行中输出两文件的相似度，即两文件的公共词汇量占两文件总词汇量的百分比，精确到小数点后1位。注意这里的一个“单词”只包括仅由英文字母组成的、长度不小于3、且不超过10的英文单词，长度超过10的只考虑前10个字母。单词间以任何非英文字母隔开。另外，大小写不同的同一单词被认为是相同的单词，例如“You”和“you”是同一个单词。

### 输入样例:

3

Aaa Bbb Ccc

#

Bbb Ccc Ddd

#

Aaa2 ccc Eee

is at Ddd@Fff

#

2

1 2

1 3

### 输出样例:

50.0%

33.3%

# 7-45 航空公司VIP客户查询（25 分）

不少航空公司都会提供优惠的会员服务，当某顾客飞行里程累积达到一定数量后，可以使用里程积分直接兑换奖励机票或奖励升舱等服务。现给定某航空公司全体会员的飞行记录，要求实现根据身份证号码快速查询会员里程积分的功能。

### 输入格式:

输入首先给出两个正整数*N*（≤10​5​​）和*K*（≤500）。其中*K*是最低里程，即为照顾乘坐短程航班的会员,航空公司还会将航程低于*K*公里的航班也按*K*公里累积。随后*N*行，每行给出一条飞行记录。飞行记录的输入格式为：18位身份证号码（空格）飞行里程。其中身份证号码由17位数字加最后一位校验码组成，校验码的取值范围为0~9和x共11个符号；飞行里程单位为公里，是(0, 15 000]区间内的整数。然后给出一个正整数*M*（≤10​5​​），随后给出*M*行查询人的身份证号码。

### 输出格式:

对每个查询人，给出其当前的里程累积值。如果该人不是会员，则输出No Info。每个查询结果占一行。

### 输入样例:

4 500

330106199010080419 499

110108198403100012 15000

120104195510156021 800

330106199010080419 1

4

120104195510156021

110108198403100012

330106199010080419

33010619901008041x

### 输出样例:

800

15000

1000

No Info

# 7-46 新浪微博热门话题（30 分）

新浪微博可以在发言中嵌入“话题”，即将发言中的话题文字写在一对“#”之间，就可以生成话题链接，点击链接可以看到有多少人在跟自己讨论相同或者相似的话题。新浪微博还会随时更新热门话题列表，并将最热门的话题放在醒目的位置推荐大家关注。

本题目要求实现一个简化的热门话题推荐功能，从大量英文（因为中文分词处理比较麻烦）微博中解析出话题，找出被最多条微博提到的话题。

### 输入格式:

输入说明：输入首先给出一个正整数*N*（≤10​5​​），随后*N*行，每行给出一条英文微博，其长度不超过140个字符。任何包含在一对最近的#中的内容均被认为是一个话题，如果长度超过40个字符，则只保留前40个字符。输入保证#成对出现。

### 输出格式:

第一行输出被最多条微博提到的话题，第二行输出其被提到的微博条数。如果这样的话题不唯一，则输出按字母序最小的话题，并在第三行输出And k more ...，其中k是另外几条热门话题的条数。输入保证至少存在一条话题。

注意：两条话题被认为是相同的，如果在去掉所有非英文字母和数字的符号、并忽略大小写区别后，它们是相同的字符串；同时它们有完全相同的分词。输出时除首字母大写外，只保留小写英文字母和数字，并用一个空格分隔原文中的单词。

### 输入样例:

4

This is a #test of topic#.

Another #Test of topic.#

This is a #Hot# #Hot# topic

Another #hot!# #Hot# topic

### 输出样例:

Hot

2

And 1 more ...

# 7-47 打印选课学生名单（25 分）

假设全校有最多40000名学生和最多2500门课程。现给出每个学生的选课清单，要求输出每门课的选课学生名单。

### 输入格式:

输入的第一行是两个正整数：N（≤40000），为全校学生总数；K（≤2500），为总课程数。此后N行，每行包括一个学生姓名（3个大写英文字母+1位数字）、一个正整数C（≤20）代表该生所选的课程门数、随后是C个课程编号。简单起见，课程从1到K编号。

### 输出格式:

顺序输出课程1到K的选课学生名单。格式为：对每一门课，首先在一行中输出课程编号和选课学生总数（之间用空格分隔），之后在第二行按字典序输出学生名单，每个学生名字占一行。

### 输入样例:

10 5

ZOE1 2 4 5

ANN0 3 5 2 1

BOB5 5 3 4 2 1 5

JOE4 1 2

JAY9 4 1 2 5 4

FRA8 3 4 2 5

DON2 2 4 5

AMY7 1 5

KAT3 3 5 4 2

LOR6 4 2 4 1 5

### 输出样例:

1 4

ANN0

BOB5

JAY9

LOR6

2 7

ANN0

BOB5

FRA8

JAY9

JOE4

KAT3

LOR6

3 1

BOB5

4 7

BOB5

DON2

FRA8

JAY9

KAT3

LOR6

ZOE1

5 9

AMY7

ANN0

BOB5

DON2

FRA8

JAY9

KAT3

LOR6

ZOE1

# 7-48 银行排队问题之单窗口“夹塞”版（30 分）

排队“夹塞”是引起大家强烈不满的行为，但是这种现象时常存在。在银行的单窗口排队问题中，假设银行只有1个窗口提供服务，所有顾客按到达时间排成一条长龙。当窗口空闲时，下一位顾客即去该窗口处理事务。此时如果已知第*i*位顾客与排在后面的第*j*位顾客是好朋友，并且愿意替朋友办理事务的话，那么第*i*位顾客的事务处理时间就是自己的事务加朋友的事务所耗时间的总和。在这种情况下，顾客的等待时间就可能被影响。假设所有人到达银行时，若没有空窗口，都会请求排在最前面的朋友帮忙（包括正在窗口接受服务的朋友）；当有不止一位朋友请求某位顾客帮忙时，该顾客会根据自己朋友请求的顺序来依次处理事务。试编写程序模拟这种现象，并计算顾客的平均等待时间。

### 输入格式:

输入的第一行是两个整数：1≤*N*≤10000，为顾客总数；0≤*M*≤100，为彼此不相交的朋友圈子个数。若*M*非0，则此后*M*行，每行先给出正整数2≤*L*≤100，代表该圈子里朋友的总数，随后给出该朋友圈里的*L*位朋友的名字。名字由3个大写英文字母组成，名字间用1个空格分隔。最后*N*行给出*N*位顾客的姓名、到达时间*T*和事务处理时间*P*（以分钟为单位），之间用1个空格分隔。简单起见，这里假设顾客信息是按照到达时间先后顺序给出的（有并列时间的按照给出顺序排队），并且假设每个事务最多占用窗口服务60分钟（如果超过则按60分钟计算）。

### 输出格式:

按顾客接受服务的顺序输出顾客名字，每个名字占1行。最后一行输出所有顾客的平均等待时间，保留到小数点后1位。

### 输入样例:

6 2

3 ANN BOB JOE

2 JIM ZOE

JIM 0 20

BOB 0 15

ANN 0 30

AMY 0 2

ZOE 1 61

JOE 3 10

### 输出样例:

JIM

ZOE

BOB

ANN

JOE

AMY

75.2

# 7-49 打印学生选课清单（25 分）

假设全校有最多40000名学生和最多2500门课程。现给出每门课的选课学生名单，要求输出每个前来查询的学生的选课清单。

### 输入格式:

输入的第一行是两个正整数：N（≤40000），为前来查询课表的学生总数；K（≤2500），为总课程数。此后顺序给出课程1到K的选课学生名单。格式为：对每一门课，首先在一行中输出课程编号（简单起见，课程从1到K编号）和选课学生总数（之间用空格分隔），之后在第二行给出学生名单，相邻两个学生名字用1个空格分隔。学生姓名由3个大写英文字母+1位数字组成。选课信息之后，在一行内给出了N个前来查询课表的学生的名字，相邻两个学生名字用1个空格分隔。

### 输出格式:

对每位前来查询课表的学生，首先输出其名字，随后在同一行中输出一个正整数C，代表该生所选的课程门数，随后按递增顺序输出C个课程的编号。相邻数据用1个空格分隔，注意行末不能输出多余空格。

### 输入样例:

10 5

1 4

ANN0 BOB5 JAY9 LOR6

2 7

ANN0 BOB5 FRA8 JAY9 JOE4 KAT3 LOR6

3 1

BOB5

4 7

BOB5 DON2 FRA8 JAY9 KAT3 LOR6 ZOE1

5 9

AMY7 ANN0 BOB5 DON2 FRA8 JAY9 KAT3 LOR6 ZOE1

ZOE1 ANN0 BOB5 JOE4 JAY9 FRA8 DON2 AMY7 KAT3 LOR6

### 输出样例:

ZOE1 2 4 5

ANN0 3 1 2 5

BOB5 5 1 2 3 4 5

JOE4 1 2

JAY9 4 1 2 4 5

FRA8 3 2 4 5

DON2 2 4 5

AMY7 1 5

KAT3 3 2 4 5

LOR6 4 1 2 4 5

# 7-50 畅通工程之局部最小花费问题（35 分）

某地区经过对城镇交通状况的调查，得到现有城镇间快速道路的统计数据，并提出“畅通工程”的目标：使整个地区任何两个城镇间都可以实现快速交通（但不一定有直接的快速道路相连，只要互相间接通过快速路可达即可）。现得到城镇道路统计表，表中列出了任意两城镇间修建快速路的费用，以及该道路是否已经修通的状态。现请你编写程序，计算出全地区畅通需要的最低成本。

### 输入格式:

输入的第一行给出村庄数目*N* (1≤*N*≤100)；随后的*N*(*N*−1)/2行对应村庄间道路的成本及修建状态：每行给出4个正整数，分别是两个村庄的编号（从1编号到*N*），此两村庄间道路的成本，以及修建状态 — 1表示已建，0表示未建。

### 输出格式:

输出全省畅通需要的最低成本。

### 输入样例:

4

1 2 1 1

1 3 4 0

1 4 1 1

2 3 3 0

2 4 2 1

3 4 5 0

### 输出样例:

3

# 7-51 两个有序链表序列的合并（20 分）

已知两个非降序链表序列S1与S2，设计函数构造出S1与S2的并集新非降序链表S3。

### 输入格式:

输入分两行，分别在每行给出由若干个正整数构成的非降序序列，用−1表示序列的结尾（−1不属于这个序列）。数字用空格间隔。

### 输出格式:

在一行中输出合并后新的非降序链表，数字间用空格分开，结尾不能有多余空格；若新链表为空，输出NULL。

### 输入样例:

1 3 5 -1

2 4 6 8 10 -1

### 输出样例:

1 2 3 4 5 6 8 10

# 7-52 两个有序链表序列的交集（20 分）

已知两个非降序链表序列S1与S2，设计函数构造出S1与S2的交集新链表S3。

### 输入格式:

输入分两行，分别在每行给出由若干个正整数构成的非降序序列，用−1表示序列的结尾（−1不属于这个序列）。数字用空格间隔。

### 输出格式:

在一行中输出两个输入序列的交集序列，数字间用空格分开，结尾不能有多余空格；若新链表为空，输出NULL。

### 输入样例:

1 2 5 -1

2 4 5 8 10 -1

### 输出样例:

2 5

# 7-53 两个有序序列的中位数（25 分）

已知有两个等长的非降序序列S1, S2, 设计函数求S1与S2并集的中位数。有序序列*A*​0​​,*A*​1​​,⋯,*A*​*N*−1​​的中位数指*A*​(*N*−1)/2​​的值,即第⌊(*N*+1)/2⌋个数（*A*​0​​为第1个数）。

### 输入格式:

输入分三行。第一行给出序列的公共长度N（0<N≤100000），随后每行输入一个序列的信息，即N个非降序排列的整数。数字用空格间隔。

### 输出格式:

在一行中输出两个输入序列的并集序列的中位数。

### 输入样例1:

5

1 3 5 7 9

2 3 4 5 6

### 输出样例1:

4

### 输入样例2:

6

-100 -10 1 1 1 1

-50 0 2 3 4 5

### 输出样例2:

1