這次作業檢查方式在hdu在線題庫：http://acm.hdu.edu.cn/，在WEB—DIY搜索"吉首大学软件服务外包学院实验室"完成杭电平台上的2012级练习以及此word中习题。

杭电习题密碼：syscontest

题目中标注的页码是《数据结构（严蔚敏）》一书

杭电有26题

附书练习如下:

一、链表部分

1、P39页-表达式加：

Sample Input

A = 2.60 - 43.10x^2

B = -77.10x^8

Sample Output

AddPolyn(A,B);

A = 2.60 - 43.10x^2 - 77.10x^8

2、表达式相乘

Sample Input

A = 2.60 - 43.10x^2

B = -77.10x^8

Sample Output

MulPolyn(A,B);

A = -200.46x^8 + 3323.01x^10

3、表达式相除

Sample Input

A = 2.60 - 43.10x^2

B = -77.10x^8

Sample Output

DivPolyn(A,B);

A = 0.00

B = 2.60 - 43.10x^2

4、合并链表

Input

输入包含四行，第一行为自然数n，表示第一个非递减顺序线性表的长度，第二行为n个自然数构成的非递减顺序线性表，第三行为自然数m，表示第二个非递减顺序线性表的长度，第四行为m个自然数构成的非递减顺序线性表。

Output

用一行输出合并后的非递减顺序线性表，各数之间用一个空格隔开。

Sample Input

2

1 3

3

2 3 6

Sample Output

1 2 3 3 6

5、链表的插入删除

Description

删除链式线性表指定位置的元素。

Input

第一行为自然数n，表示链式线性表的长度，第二行为n个自然数表示链式线性表各元素值。第三行为指定删除的位置，第四行为插入的位置及元素值（如果位置不对，则不作操作，插入位置应在删除元素后重新计数）。

Output

输出删除与插入元素后的链式线性表的所有元素，元素之间用一个空格隔开。

Sample Input

5 1 2 3 4 5

3

6 7

Sample Output

1 2 4 5

6、链表实现数据的有序合并

Input

输入有多组测试数据，每组测试数据包括2行，每个链表用一行表示。

Output

输出合并后链表并按从小到大顺序进行排序和输出其内容。

Sample Input

List A = ( 18 40 44 52 53 59 )

List B = ( 15 18 19 25 27 35 48 50 55 )

List A = ( 1 7 26 27 29 41 45 47 56 )

List B = ( 3 11 13 22 22 33 34 44 46 49 57 )

Sample Output

MergeList\_L(A, B, C); List C = ( 15 18 18 19 25 27 35 40 44 48 50 52 53 55 59 )

MergeList\_L(A, B, C); List C = ( 1 3 7 11 13 22 22 26 27 29 33 34 41 44 45 46 47 49 56 57 )

Ps:希望大家通过练习熟练掌握链表的各种操作（插入，删除，排序，合并……）以及掌握单向链表，双向链表，循环链表结构。

二、栈和队列

1、进制的转换

将输入的十进制数转换为二进制

Sample Input

2

0

12

1

Sample Output

2―->10

0-->0

12-->1100

1-->1

2、行编辑程序

如果遇到‘#’，表示后退一格，即前一字符无效，如果遇到@，表示前一单词无效，即退出到空格或所在行头为止。采用栈实现。

Sample Input

whli##ilr#e(s# \*s)

outcha@putchar( \*s =# ++)

Sample Output

while( \*s)

putchar( \*s ++)

3、括号匹配的判断

Sample Input

2

[([][]())]

)[]()

Sample Output

YES

NO

4、迷宫求解p51

Sample input

1

11111

10101

10101

10001

11111

Sample output

Yes

5、表达式求值

Description

利用栈来实现含有加，减，乘，除等基本运算，输出表达式的值

Input

3x(15/5)+8

Output

17

Sample Input

24-[6+(27/3)x2]

Sample Output

0

6、栈和递归的实现汉诺塔

Description

有A、B、C三个杆，A上放着N的大小不等的盘子，怎样将A上的所有盘子放到C上，移动过程中，每次只能移动一个盘子且始终保持三个杆上的大盘子在下，小盘子在上

Input

A杆上的盘子数

Output

盘子移动的次数

Sample Input

5

Sample Output

31

7、用队列实现1题的进制转换

8、简单出队练习

Description

排队是我们日常生活中最常见的事情。现有若干个人在银行要求处理各种存取款业务，采用先来先到的顺序进行处理。假定“IN”表示到达银行，“OUT”表示进行处理。要求输出所有顾客的处理顺序。

Input

输 入第一行包括一个整数n(1<=n<=10000000)表示有多少处理情况，接下来n行，每行以“IN”或“OUT”开头，如果是“IN” 开头，表示达到银行并排队，后面接着为某顾客的姓名，用一个音词表示姓名，用一空格隔开，如果是“OUT”，表示从队列中出一个人到柜台处理业务。

Output

输出所有按先后顺序到柜台处理业务的顾客姓名，每行为一顾客。

Sample Input

5

IN bob

IN Ruse

OUT

IN SanLU

OUT

Sample Output

Bob

Ruse

9、火车重排p61

Input

一个数n(1< n< 27)及由n个小写字母组成的字符串。

Output

可以调度则输出最短的调度序列，不可以调度时则输出‘NO’。

Sample Input

3

cba

Sample Output

c A B

b A C

a A D

b C D

c B D

ps：站和队列具体掌握栈的出栈规律，理解栈和递归的原理；队列掌握出队入队规律，重点掌握循环队列出队入队的处理。

三、串

1、kmp算法p79

Description

输入一个主串和一个子串，若匹配成功，则找出匹配的趟数和在子串在主串中的位置，若匹配不成功，则输出0

Input

输入两个字符串

Output

输出匹配的趟数和位置

Sample Input

Ababcabcacbab

abcac

Sample Output

3 6

Ps：理解模式匹配的原理

四、树和二叉树

1、遍历二叉树

Description

给定一个序列，按先序序列建立二叉树。输出建立后的二叉树的从上到下层次遍历序列、先序遍历结果、中序遍历结果、后续遍历结果

Input

一个序列

Output

从上之下层次序列

Sample Input

ABC##DE#G##F###

Sample Output

ABCDEFG

ABCDEGF

CBEGDFA

CGEFDBA

2、哈夫曼编码

Description

编写一个程序实现构造一颗哈夫曼树的功能和给出哈夫曼编码的编码功能。其中要求，哈夫曼树是自左到右按结点的权值从小到大的排列，如果结点的权值相等时，非终端结点优先与终端结点，即终端结点排在右边，如果都是同类型的结点，则按输入的顺序和子结点的顺序自左到右排列。

Input

输入包括多组测试数据，每组测试数据包括多个字符和相应的权值，

Output

计算出每个字符的哈夫曼编码并输出。

Sample Input

A(5) B(29) C(7) D(8) E(14) F(23) G(3) H(11)

Sample Output

A( 5) : 0001

B(29) : 10

C( 7) : 1110

D( 8) : 1111

E(14) : 110

F(23) : 01

G( 3) : 0000

H(11) : 001

3、中序后序 求先序

Description

已知二叉树的中序、后序，建树求先序。

Input

中序序列

后序序列

Output

先序序列

Sample Input

d g b a e c f

g d b e f c a

Sample Output

a b d g c e f

4、中序先序 求后序

Description

已知二叉树的中序、后序，建树求先序。

Input

中序序列

先序序列

Output

后续序列

Sample Input

d g b a e c f

a b d g c e f

Sample Output

g d b e f c a

5、二叉排序树

Description

二叉排序树的定义是：或者是 一棵空树，或者是具有下列性质的二叉树：（1）若它的左子树不空，则左子树上所有的结点值均小于它的根结点的值；（2）若它的右子树不空，则右子树上所有 结点的值均大于或等于它的根结点的值；（3）它的左右子树也分别为二叉排序树。现要求根据输入的元素值，构造一棵二叉排序树，并输出其先序遍历、中序遍历 和后序遍历结果。

Input

输入第一行为测试用例个数n，接下来为n个测试用例，每个测试用例占两行，其中第一行为元素个数m，第二行为m个需要构造成二叉排序树的元素值。

Output

每个测试用例用三行输出，其中第一行输出先序遍历结果，第二行输出中序遍历结果，第三行输出后序遍历结果。各元素之间用一个空格隔开。

Sample Input

1

5

8 4 2 7 4

Sample Output

8 4 2 7 4

2 4 4 7 8

2 4 7 4 8

6、应用FBI树

Description

  我们可以把由“0”和“1”组成的字符串分为三类：全“0”串称为B串，全“1”串称为I串，既含“0”又含“1”的串则称为F串。         FBI树是一种二叉树1，它的结点类型也包括F结点，B结点和I结点三种。由一个长度为2^N的“01”串S可以构造出一棵FBI树T，递归的构造方法如 下：         1)  T的根结点为R，其类型与串S的类型相同；         2)  若串S的长度大于1，将串S从中间分开，分为等长的左右子串S1和S2；由左子串S1构造R的左子树T1，由右子串S2构造R的右子树T2。         现在给定一个长度为2^N的“01”串，请用上述构造方法构造出一棵FBI树，并输出它的后序遍历2序列。

Input

输入的第一行是一个整数N(0< =N< =10)，第二行是一个长度为2^N的“01”串。

Output

  输出包括一行，这一行只包含一个字符串，即FBI树的后序遍历序列。

Sample Input

3

10001011

Sample Output

IBFBBBFIBFIIIFF

PS：通过树的学习重点理解递归的思想

五、图

1、DFS

Description

图(graph)是数据结构 G=(V,E)，其中V是G中结点的有限非空集合,结点的偶对称为边(edge)；E是G中边的有限集合。设V={0,1,2,……,n-1}，图中的结点又称为顶点(vertex)，有向图(directed graph)指图中代表边的偶对是有序的，用<u，v>代表一条有向边（又称为弧），则u称为该边的始点（尾），v称为边的终点（头）。无向图(undirected graph)指图中代表边的偶对是无序的，在无向图中边(u，v )和(v，u)是同一条边。

输入边构成无向图，求以顶点0为起点的深度优先遍历序列。

Input

第一行为两个整数n、e，表示图顶点数和边数。以下e行每行两个整数，表示一条边的起点、终点，保证不重复、不失败。1≤n≤20，0≤e≤190

（有多组测试数据）

Output

前面n行输出无向图的邻接矩阵，最后一行输出以顶点0为起点的深度优先遍历序列，对于任一起点，首先遍历的是终点序号最小的、尚未被访问的一条边。每个序号后输出一个空格。每组结果换行

Sample Input

4 5

0 1

0 3

1 2

1 3

2 3

Sample Output

0 1 0 1

1 0 1 1

0 1 0 1

1 1 1 0

0 1 2 3

2、BFS

Description

图(graph)是数据结构 G=(V,E)，其中V是G中结点的有限非空集合,结点的偶对称为边(edge)；E是G中边的有限集合。设V={0,1,2,……,n-1}，图中的结点又称为顶点(vertex)，有向图(directed graph)指图中代表边的偶对是有序的，用<u，v>代表一条有向边（又称为弧），则u称为该边的始点（尾），v称为边的终点（头）。无向图(undirected graph)指图中代表边的偶对是无序的，在无向图中边(u，v )和(v，u)是同一条边。

输入边构成无向图，求以顶点0为起点的宽度优先遍历序列。

Input

第一行为两个整数n、e，表示图顶点数和边数。以下e行,每行两个整数，表示一条边的起点、终点，保证不重复、不失败。1≤n≤20，0≤e≤190

Output

前面n行输出无向图的邻接矩阵，最后一行输出以顶点0为起点的广度优先遍历序列，对于任一起点，按终点序号从小到大的次序遍历每一条边。每个序号后输出一个空格。

Sample Input

4 5

0 1

0 3

1 2

1 3

2 3

Sample Output

0 1 0 1

1 0 1 1

0 1 0 1

1 1 1 0

0 1 3 2

Ps：理解DFS 和 BFS