Problem Description

|  |
| --- |
| 一个长度为len（1<= len <=1000000）的顺序表，数据元素的类型为整型，将该表分为两半，前一半有m个元素，后一半有len-m个元素（1<= m <= len），设计一个时间复杂度为O（N），空间复杂度为O（1）的算法，改变原来的顺序表，把顺序表中原来在前的m个元素放到表的后段，后len-m个元素放到表的前段。    注意：先将顺序表元素调整为符合要求的内容后，再做输出，输出过程只能用一个循环语句实现，不能分成两个部分。 |

Input

|  |
| --- |
| 第一行输入整数n，代表下面有n行输入；  之后输入n行，每行先输入整数len与整数m（分别代表本表的元素总数与前半表的元素个数），之后输入len个整数，代表对应顺序表的每个元素。 |

Output

|  |
| --- |
| 输出有n行，为每个顺序表前m个元素与后（len-m）个元素交换后的结果 |

Example Input

|  |
| --- |
| 2  10 3 1 2 3 4 5 6 7 8 9 10  5 3 10 30 20 50 80 |

Example Output

|  |
| --- |
| 4 5 6 7 8 9 10 1 2 3  50 80 10 30 20 |

Problem Description

|  |
| --- |
| 已知顺序表A与B是两个有序的顺序表，其中存放的数据元素皆为普通整型，将A与B表归并为C表，要求C表包含了A、B表里所有元素，并且C表仍然保持有序。 |

Input

|  |
| --- |
| 输入分为三行：  第一行输入m、n（1<= m,n <=10000）的值，即为表A、B的元素个数；  第二行输入m个有序的整数，即为表A的每一个元素；  第三输入n有序的整数，即为表A的每一个元素； |

Output

|  |
| --- |
| 输出为一行，即将表A、B合并为表C后，依次输出表C所存放的元素。 |

Example Input

|  |
| --- |
| 5 3  1 3 5 6 9  2 4 10 |

Example Output

|  |
| --- |
| 1 2 3 4 5 6 9 10 |

Problem Description

|  |
| --- |
| 输入多个整数，以-1作为结束标志，顺序建立一个带头结点的单链表，之后对该单链表的数据进行逆置，并输出逆置后的单链表数据。 |

Input

|  |
| --- |
| 输入多个整数，以-1作为结束标志。 |

Output

|  |
| --- |
| 输出逆置后的单链表数据。 |

Example Input

|  |
| --- |
| 12 56 4 6 55 15 33 62 -1 |

Example Output

|  |
| --- |
| 62 33 15 55 6 4 56 12 |

Problem Description

|  |
| --- |
| 按照数据输入的相反顺序（逆位序）建立一个单链表，并将单链表中复杂的元素删除（值相同的元素之保留最后输入的一个）。 |

Input

|  |
| --- |
| 第一行输入元素个数n；  第二行输入n个整数。 |

Output

|  |
| --- |
| 第一行输入初始链表元素个数；  第二行输出按照逆位序所建立的初始链表;  第三行输出删除重复元素后的单链表元素个数；  第四行输出删除重复元素后的单链表。 |

Example Input

|  |
| --- |
| 10  21 30 14 55 32 63 11 30 55 30 |

Example Output

|  |
| --- |
| 10  30 55 30 11 63 32 55 14 30 21  7  30 55 11 63 32 14 21 |

Problem Description

|  |
| --- |
| 输入一个十进制整数，将其转换成对应的R（2<= R <= 9）进制数，并输出。 |

Input

|  |
| --- |
| 第一行输入需要转换的十进制数；  第二行输入R。 |

Output

|  |
| --- |
| 输出转换所得的R进制数。 |

Example Input

|  |
| --- |
| 1279  8 |

Example Output

|  |
| --- |
| 2377 |

Problem Description

|  |
| --- |
| 给你一串字符，不超过50个字符，可能包括括号、数字、字母、标点符号、空格，你的任务是检查这一串字符中的（），[],{}是否匹配。 |

Input

|  |
| --- |
| 输入数据有多组，处理到文件结束。 |

Output

|  |
| --- |
| 如果匹配就输出“yes”，不匹配输出“no” |

Example Input

|  |
| --- |
| Sin（20+10）  {[}] |

Example Output

|  |
| --- |
| Yes  No |

Problem Description

|  |
| --- |
| 一个简单的行编辑程序的功能是：接受用户从终端输入的程序或数据，并存入用户的数据区。  由于用户在终端上进行输入时，不能保证不出差错，因此，若在编辑程序中，“每接受一个字符即存入用户数据区”的做法显然不是最恰当的。最好的做法是，设立一个输入缓冲区，用以接受用户输入的一行字符，然后逐行存入用户数据区。允许用户输入出差错，并在发现有误时可以及时更正。例如，当互用发现刚刚键入的一个字符是错的时，可补进一个退格符“#”，以表示前一个字符无效；  如果发现当时键入的行内差错较多或难以补救，则可以键入一个退行符“@”，以表示当时行中的字符均无效。  如果已经在行首继续输入‘#’符号无效。 |

Input

|  |
| --- |
| 输入多行字符序列，行字符总数（包含退格符和退行符）不大于250. |

Output

|  |
| --- |
| 按照上述说明得到的输出。 |

Example Input

|  |
| --- |
| whli##ilr#e（s#\*s）  outcha@putchar（\*s=#++）； |

Example Output

|  |
| --- |
| while(\*s)  putchar(\*s++)； |

Problem Description

|  |
| --- |
| 已知一棵二叉树的前序遍历和中序遍历，求二叉树的后序遍历。 |

Input

|  |
| --- |
| 输入数据有多组，第一行是一个整数t（t<1000），代表有t组测试数据，每组包括两个长度小于50的字符串，第一个字符串表示二叉树的先序遍历序列，第二个字符串表示二叉树的中序遍历序列。 |

Output

|  |
| --- |
| 每组第一行输出二叉树的后序遍历序列，第二行输出二叉树的层次遍历序列。 |

Example Input

|  |
| --- |
| 2  abdegcf  dbgeafc  xnliu  lnixu |

Example Output

|  |
| --- |
| dgebfca  abcdefg  linux  xnuli |

Problem Description

|  |
| --- |
| 已知一个按先序输入的字符序列，如abd，，eg，，，cf，，，（其中，表示空结点。）请建立该二叉树并按从上到下从左到右的顺序输出该二叉树的所有叶子结点。 |

Input

|  |
| --- |
| 输入数据有多行，每一行是一个长度小于50个字符的字符串。 |

Output

|  |
| --- |
| 按从上到下从左到右的顺序输出二叉树的叶子结点。 |

Example Input

|  |
| --- |
| abd，，eg，，，cf，，，  xnl，，i，，u，， |

Example Output

|  |
| --- |
| dfg  uli |

Problem Description

|  |
| --- |
| 作为一个资深驴友，小新有一张珍藏的自驾旅游线路图，图上详细的标注了全国各个城市之间的高速公路距离和公路收费情况，现在请你编写一个程序，找出一条出发地到目的地之间的最短路径，如果有多条路径最短，则输出过路费最少的一条路径。 |

Input

|  |
| --- |
| 连续T组数据输入，每组输入数据的第一行给出四个正整数N、M，s，d，其中N（2<= N <=500）是城市数目，城市标号从0～N-1，M是城市间高速公路的条数，s是出发地的城市编号，d是目的地的城市编号了随后M行，每行给出一条高速公路的信息，表示城市1、城市2、高速公路长度、收费额，中间以空格间隔，数字均为整数且不超过500，输入数据均保证有解。 |

Output

|  |
| --- |
| 在同一行中输出路径长度和收费总额，数据间用空格间隔。 |

Example Input

|  |
| --- |
| 1  4 5 0 3  0 1 1 20  1 3 2 30  0 3 4 10  0 2 2 20  2 3 1 20 |

Example Output

|  |
| --- |
| 3 40 |