**排序类问题**

**1151 成绩排序**

Time Limit: 1000 ms

Memory Limit: 32768 mb

输入任意（用户，成绩）序列，可以获得成绩从高到低或从低到高的排列,相同成绩

都按先录入排列在前的规则处理。

示例：

jack 70

peter 96

Tom 70

smith 67

从高到低 成绩

peter 96

jack 70

Tom 70

smith 67

从低到高

smith 67

jack 70

Tom 70

peter 96

**输入输出格式**

**输入描述:**

输入多行，先输入要排序的人的个数，然后输入排序方法0（降序）或者1（升序）再分别输入他们的名字和成绩，以一个空格隔开

**输出描述:**

按照指定方式输出名字和成绩，名字和成绩之间以一个空格隔开

**输入输出样例**

**输入样例#:**

3

0

fang 90

yang 50

ning 70

**输出样例#:**

fang 90

ning 70

yang 50

**题目来源**

**清华大学上机题**

**1010 排序**

Time Limit: 1000 ms

Memory Limit: 256 mb

输入n个数进行排序，要求先按奇偶后按从小到大的顺序排序。

**输入输出格式**

**输入描述:**

第一行输入一个整数n，表示总共有多少个数，n<=1000。

第二行输入n个整数，用空格隔开。

**输出描述:**

输出排序之后的结果。

**输入输出样例**

**输入样例#:**

8

1 2 3 4 5 6 7 8

**输出样例#:**

1 3 5 7 2 4 6 8

**题目来源**

**贵州大学2018/兰州大学2018年机试**

**1106 排序2**

Time Limit: 1000 ms

Memory Limit: 256 mb

编写程序实现直接插入排序、希尔排序(d=5)(d=5即增量,例如对i,i+5,i+10排序)、直接选择排序、快速排序和二路归并排序算法。

**输入输出格式**

**输入描述:**

第一行是待排序数据元素的个数，第二行是待排序的数据元素。

**输出描述:**

直接插入排序后的结果

一趟希尔排序后的结果

直接选择排序后的结果

快速排序后的结果

一趟二路归并排序后的结果

**输入输出样例**

**输入样例#:**

10

50 36 41 19 23 4 20 18 12 22

**输出样例#:**

4 12 18 19 20 22 23 36 41 50

4 20 18 12 22 50 36 41 19 23

4 12 18 19 20 22 23 36 41 50

4 12 18 19 20 22 23 36 41 50

36 50 19 41 4 23 18 20 12 22**​**

**1159 成绩排序2.0**

Time Limit: 1000 ms

Memory Limit: 32768 mb

用一维数组存储学号和成绩，然后，按成绩排序输出。

**输入输出格式**

**输入描述:**

输入第一行包括一个整数N(1<=N<=100)，代表学生的个数。

接下来的N行每行包括两个整数p和q，分别代表每个学生的学号和成绩。

**输出描述:**

按照学生的成绩从小到大进行排序，并将排序后的学生信息打印出来。

如果学生的成绩相同，则按照学号的大小进行从小到大排序。

**输入输出样例**

**输入样例#:**

3

1 90

2 87

3 92

**输出样例#:**

2 87

1 90

3 92

**题目来源**

**清华大学上机题**

**1217 国名排序**

Time Limit: 1000 ms

Memory Limit: 256 mb

小李在准备明天的广交会，明天有来自世界各国的客户跟他们谈生意，小李要尽快的整理出名单给经理，你能帮他把客户来自的国家按英文字典次序排好吗？例如小李手上有来自加拿大，美国，中国的名单，排好的名单应是美国，加拿大，中国

**输入输出格式**

**输入描述:**

第一行为一个n（n<=100）表示n个国家，第２行到第n＋１行分别为n个国家的名字．

**输出描述:**

输出n个国家按字典顺序的排列．

**输入输出样例**

**输入样例#:**

3

China

Canada

America

**输出样例#:**

America

Canada

China

**1227 日志排序**

Time Limit: 1000 ms

Memory Limit: 256 mb

有一个网络日志，记录了网络中计算任务的执行情况，每个计算任务对应一条如下形式的日志记录： “hs\_10000\_p”是计算任务的名称， “2007-01-17 19:22:53,315”是计算任务开始执行的时间“年-月-日 时：分：秒，毫秒”， “253.035(s)”是计算任务消耗的时间(以秒计) hs\_10000\_p 2007-01-17 19:22:53,315 253.035(s) 请你写一个程序，对日志中记录的计算任务进行排序。 时间消耗少的计算任务排在前面，时间消耗多的计算任务排在后面。 如果两个计算任务消耗的时间相同，则将开始执行时间早的计算任务排在前面。

**输入输出格式**

**输入描述:**

日志中每个记录是一个字符串，每个字符串占一行。最后一行为空行，表示日志结束。日志中最多可能有10000条记录。

计算任务名称的长度不超过10，开始执行时间的格式是YYYY-MM-DD HH:MM:SS,MMM，消耗时间小数点后有三位数字。

计算任务名称与任务开始时间、消耗时间之间以一个或多个空格隔开，行首和行尾可能有多余的空格。

**输出描述:**

排序好的日志记录。每个记录的字符串各占一行。

输入的格式与输入保持一致，输入包括几个空格，你的输出中也应该包含同样多的空格。

**输入输出样例**

**输入样例#:**

hs\_10000\_p 2007-01-17 19:22:53,315 253.035(s)

hs\_10001\_p 2007-01-17 19:22:53,315 253.846(s)

hs\_10002\_m 2007-01-17 19:22:53,315 129.574(s)

hs\_10002\_p 2007-01-17 19:22:53,315 262.531(s)

hs\_10003\_m 2007-01-17 19:22:53,318 126.622(s)

hs\_10003\_p 2007-01-17 19:22:53,318 136.962(s)

hs\_10005\_m 2007-01-17 19:22:53,318 130.487(s)

hs\_10005\_p 2007-01-17 19:22:53,318 253.035(s)

hs\_10006\_m 2007-01-17 19:22:53,318 248.548(s)

hs\_10006\_p 2007-01-17 19:25:23,367 3146.827(s)

**输出样例#:**

hs\_10003\_m 2007-01-17 19:22:53,318 126.622(s)

hs\_10002\_m 2007-01-17 19:22:53,315 129.574(s)

hs\_10005\_m 2007-01-17 19:22:53,318 130.487(s)

hs\_10003\_p 2007-01-17 19:22:53,318 136.962(s)

hs\_10006\_m 2007-01-17 19:22:53,318 248.548(s)

hs\_10000\_p 2007-01-17 19:22:53,315 253.035(s)

hs\_10005\_p 2007-01-17 19:22:53,318 253.035(s)

hs\_10001\_p 2007-01-17 19:22:53,315 253.846(s)

hs\_10002\_p 2007-01-17 19:22:53,315 262.531(s)

hs\_10006\_p 2007-01-17 19:25:23,367 3146.827(s)

**题目来源**

**北京大学机考题**

**1248 整数奇偶排序**

Time Limit: 1000 ms

Memory Limit: 256 mb

输入10个整数，彼此以空格分隔。重新排序以后输出(也按空格分隔)，要求: 1.先输出其中的奇数,并按从大到小排列； 2.然后输出其中的偶数,并按从小到大排列。

**输入输出格式**

**输入描述:**

任意顺序的10个整数（0～100），彼此以空格分隔。

**输出描述:**

可能有多组测试数据，对于每组数据，按照要求排序后输出，由空格分隔。

测试数据可能有很多组，请使用while(cin>>a[0]>>a[1]>>...>>a[9])类似的做法来实现;

输入数据随机，有可能相等。

**输入输出样例**

**输入样例#:**

4 7 3 13 11 12 0 47 34 98

**输出样例#:**

47 13 11 7 3 0 4 12 34 98

**题目来源**

**北京大学机试题**

**1254 字符串排序**

Time Limit: 1000 ms

Memory Limit: 256 mb

输入一个长度不超过20的字符串，对所输入的字符串，按照ASCII码的大小从小到大进行排序，请输出排序后的结果

**输入输出格式**

**输入描述:**

一个字符串，其长度n<=20

**输出描述:**

输入样例可能有多组，对于每组测试样例，按照ASCII码的大小对输入的字符串从小到大进行排序，输出排序后的结果

**输入输出样例**

**输入样例#:**

dcba

**输出样例#:**

abcd

**题目来源**

**北京大学机试题**

**1255 字符串排序2**

Time Limit: 1000 ms

Memory Limit: 256 mb

编写一个程序，将输入字符串中的字符按如下规则排序(一个测试用例可能包含多组数据，请注意处理)。

规则1 ：英文字母从A到Z排列，不区分大小写。

如，输入：Type 输出：epTy

规则2 ：同一个英文字母的大小写同时存在时，按照输入顺序排列。

如，输入：BabA 输出：aABb

规则3 ：非英文字母的其它字符保持原来的位置。

如，输入：By?e 输出：Be?y

**输入输出样例**

**输入样例#:**

A Famous Saying: Much Ado About Nothing (2012/8).

**输出样例#:**

A aaAAbc dFgghh: iimM nNn oooos Sttuuuy (2012/8).

**题目来源**

**北京大学机试题**

**1261 字符串排序3**

Time Limit: 1000 ms

Memory Limit: 256 mb

先输入你要输入的字符串的个数。然后换行输入该组字符串。每个字符串以回车结束，每个字符串少于一百个字符。 如果在输入过程中输入的一个字符串为“stop”，也结束输入。 然后将这输入的该组字符串按每个字符串的长度，由小到大排序，按排序结果输出字符串。

**输入输出格式**

**输入描述:**

字符串的个数，以及该组字符串。每个字符串以‘\n’结束。如果输入字符串为“stop”，也结束输入.

**输出描述:**

可能有多组测试数据，对于每组数据，

将输入的所有字符串按长度由小到大排序输出(如果有“stop”，不输出“stop”)。

根据输入的字符串个数来动态分配存储空间（采用new()函数）。每个字符串会少于100个字符。

测试数据有多组，注意使用while()循环输入。

**输入输出样例**

**输入样例#:**

5

sky is grey

cold

very cold

stop

3

it is good enough to be proud of

good

it is quite good

**输出样例#:**

cold

very cold

sky is grey

good

it is quite good

it is good enough to be proud of

**题目来源**

**北京大学机试题**

**1294 后缀子串排序**

Time Limit: 1000 ms

Memory Limit: 256 mb

对于一个字符串，将其后缀子串进行排序，例如grain其子串有：grain rain ain in n 然后对各子串按字典顺序排序，即：ain,grain,in,n,rain

**输入输出格式**

**输入描述:**

多组数据输入。每个案例为一行字符串。

**输出描述:**

将子串排序输出

**输入输出样例**

**输入样例#:**

grain

**输出样例#:**

ain

grain

in

n

rain

**题目来源**

**上海交通大学机试题**

**1310 奥运排序问题**

Time Limit: 1000 ms

Memory Limit: 256 mb

按要求，给国家进行排名。

**输入输出格式**

**输入描述:**

有多组数据。

第一行给出国家数N，要求排名的国家数M，国家号从0到N-1。

第二行开始的N行给定国家或地区的奥运金牌数，奖牌数，人口数（百万）。

接下来一行给出M个国家号。

**输出描述:**

排序有4种方式: 金牌总数 奖牌总数 金牌人口比例 奖牌人口比例

对每个国家给出最佳排名排名方式 和 最终排名

格式为: 排名:排名方式

如果有相同的最终排名，则输出排名方式最小的那种排名，对于排名方式，金牌总数 < 奖牌总数 < 金牌人口比例 < 奖牌人口比例

如果有并列排名的情况，即如果出现金牌总数为 100,90,90,80.则排名为1,2,2,4.

每组数据后加一个空行。

**输入输出样例**

**输入样例#:**

4 4

4 8 1

6 6 2

4 8 2

2 12 4

0 1 2 3

4 2

8 10 1

8 11 2

8 12 3

8 13 4

0 3

**输出样例#:**

1:3

1:1

2:1

1:2

1:1

1:1

**题目来源**

**浙江大学机试题**