**Problem A: Print Graphics Problerm (III) (Append Code)**

Time Limit: 1 Sec  Memory Limit: 2 MB  
Submit: 10086  Solved: 5335  
[[Submit](http://192.168.119.211/JudgeOnline/submitpage.php?cid=4641&pid=0&langmask=1022)][[Status](http://192.168.119.211/JudgeOnline/problemstatus.php?id=1143)]

**Description**

向标准输出上打印一些用ASCII字符组成的图形。

-----------------------------------------------------------------------------

编写一个函数用于输出ASCII图形。

用C语言实现：append.c中函数原型为

int print\_graphic(int n, char c);

用C++实现：append.cc中函数原型为

int printGraphic(int n, char c);

功能：输出n层有字符c组成的图形。

函数的调用格式见“Append Code”。

**Input**

输入为一个整数n和一个字符c，0<n<100。

**Output**

输出一个n层等腰三角形，由字符c组成，格式见sample。

**Sample Input**

5 \*

**Sample Output**

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

**HINT**

**Problem B: 编写函数：浮点数取整 (Append Code)**

Time Limit: 1 Sec  Memory Limit: 2 MB  
Submit: 11771  Solved: 4918  
[[Submit](http://192.168.119.211/JudgeOnline/submitpage.php?cid=4641&pid=1&langmask=1022)][[Status](http://192.168.119.211/JudgeOnline/problemstatus.php?id=1217)]

**Description**

各种程序设计语言里常见的取整函数有四个，分别是fix()、floor()、ceil()、round()。它们的功能细微之处各不相同，但有一点功能是共同的，就是把一个可能带小数点的浮点数转换成一个整数。

-----------------------------------------------------------------------------

编写函数myFloor()和myCeil()：

原型：int myFloor(double data);

功能：下取整，返回不大于data的最大整数。

原型：int myCeil(double data);

功能：上取整，返回不小于data的最小整数。

函数的调用格式见“Append Code”。

-----------------------------------------------------------------------------

Invalid Word（禁用单词）错误：在解决这个题目时，某些关键词是不允许被使用的。如果提交的程序中包含了下列的关键词之一，就会产生这个错误。

被禁用的头文件：math.h和stdlib.h。

**Input**

输入有多行，每行是一个可能带小数点的浮点数m。

**Output**

输出为多行，与上述输入一一对应。每行输出两个整数：m上取整和下取整的结果，两者之间用一个空格分隔开。

**Sample Input**

1.3

-1.5

2

**Sample Output**

1 2

-2 -1

2 2

**HINT**

**Problem C: 登录密码验证**

Time Limit: 1 Sec  Memory Limit: 2 MB  
Submit: 16662  Solved: 6143  
[[Submit](http://192.168.119.211/JudgeOnline/submitpage.php?cid=4641&pid=2&langmask=1022)][[Status](http://192.168.119.211/JudgeOnline/problemstatus.php?id=1093)]

**Description**

编写一个程序，模拟用户登录系统的密码验证过程。系统提供给用户的密码长度最长为20个字符，若密码输入错误可以再次输入。但为了保证用户密码安全，若连续输入密码错误超过5次就会锁定账号一段时间。

**Input**

输入为若干个串，至EOF结束。输入的第一个串是用户的正确密码，后面的串为模拟用户登录时的输入的密码。

**Output**

每次输入错误的密码，输出一个“Wrong!”，若输入的密码为正确的，输出一个“Welcome!”，并结束密码测试。若前5次输入的密码都是错误的，则后面的输入中不管是否有正确的密码都输出“Out of limited!”。

**Sample Input**

abcdefg

123456 kkkkkkkk abcdefg

**Sample Output**

Wrong!

Wrong!

Welcome!

**HINT**

输入可以用scanf("%s")处理，密码比较用字符串的比较可以完成。

**Problem D: 字符串的逆序**

Time Limit: 1 Sec  Memory Limit: 2 MB  
Submit: 13580  Solved: 8200  
[[Submit](http://192.168.119.211/JudgeOnline/submitpage.php?cid=4641&pid=3&langmask=1022)][[Status](http://192.168.119.211/JudgeOnline/problemstatus.php?id=1092)]

**Description**

将输入的一个字符串s逆序输出。

**Input**

输入为一个串s。输入最少为一个字符，最多不会超过100个字符。输入不含各种空白符（'\t'、'\n'、'\r'、' '）。

**Output**

串s的逆序。

**Sample Input**

abcde

**Sample Output**

edcba

**HINT**

输入可以用scanf("%s")处理。

**Problem E: 时间：24小时制转12小时制**

Time Limit: 1 Sec  Memory Limit: 2 MB  
Submit: 10918  Solved: 3548  
[[Submit](http://192.168.119.211/JudgeOnline/submitpage.php?cid=4641&pid=4&langmask=1022)][[Status](http://192.168.119.211/JudgeOnline/problemstatus.php?id=1163)]

**Description**

编写一个程序，把输入的24小时制的时间，转换12小时制的格式输出。

12小时制没有0点时段，是以数字12、1、2、3、4、5、6、7、8、9、10、11依次序表示每个时段的。

正午是“12:00 p.m.”，也就是24小时制的中午12点；

24小时制的12:00～12:59，是12小时制的12:00 p.m.～12:59 p.m.；

24小时制的13:00～23:59是十二小时制的1:00 p.m.～11:59 p.m.。

午夜是“12:00 a.m.”，也就是24小时制的0点，或者24点；

24小时制的00:00～00:59，是12小时制的12:00 a.m.～12:59 a.m.；

24小时制的1:00～11:59是十二小时制的1:00 a.m.～11:59 a.m.。

**Input**

输入为一行。输入为24小时制的小时和分，都占满2个字符的位置，用“:”分隔。范围是00:00～23:59。

**Output**

输出为12小时制的小时和分，都占满2个字符的位置，用“:”分隔，一个空格后跟“a.m.”（午前）或“p.m.”（午后）。

**Sample Input**

21:05

**Sample Output**

09:05 p.m.

**HINT**

**Problem F: 成绩统计**

Time Limit: 1 Sec  Memory Limit: 16 MB  
Submit: 8921  Solved: 4095  
[[Submit](http://192.168.119.211/JudgeOnline/submitpage.php?cid=4641&pid=5&langmask=1022)][[Status](http://192.168.119.211/JudgeOnline/problemstatus.php?id=1177)]

**Description**

把百分制的考试成绩转换为五个等级：

90～100：Excellent

80～89：Good

70～79：Average

60～69：Pass

0～59：Failing

统计每个等级的成绩，并输出。

**Input**

输入多行，每行一个整数。输入的成绩总数不超过1000个。到EOF结束。

**Output**

按从好到差的顺序分等级输出成绩，第一行为“等级 : 成绩数”，成绩数为这一等级的成绩总数；第二行为具体的成绩，用逗号分隔开，每个逗号后面有空格，每行最后一个成绩后没有逗号和空格。最后一行后有回车。输出的成绩与输入的顺序相同。若某个等级没有成绩，则不输出第二行。

具体格式参照sample。

**Sample Input**

-1

81

92

35

68

72

100

1000

95

60

59

72

**Sample Output**

Excellent : 3

92, 100, 95

Good : 1

81

Average : 2

72, 72

Pass : 2

68, 60

Failing : 2

35, 59

**HINT**

如果没有不及格的成绩，你的程序还能否正确运行吗？