The Great Computer Challenge
C ++
Level IV

Problem 1: *Sorting*

Write a C++ program that will accept a list of up to 40 data values separated by blanks, all on one line, and analyze the order of the list. Your program should output that the order of the values is either ascending or descending as appropriate and print the list of values. If the values are mixed order, your program should order the values and report the order in which it has placed the values and print the ordered set of values.

Examples:
Input: 1 2 3 4 5
Output: ascending order 1, 2, 3, 4, 5

Input: 8 7 5 3 1 -1 -7
Output: descending order 8 7 5 3 1 -1 -7

Input: 14 3 -8 2 88 4
Output: ordered the values in descending order 88, 14, 4, 3, 2, -8

Problem 2: *Wildcards*

In many computer applications it is desirable to be able to specify a pattern that can be used to identify a group of files by name. For example, if we wanted the program to deal with the files 'FNA.DOC', 'FNB.DOC', and 'FNC.DOC', we would like a way to say "get files that match the pattern FN plus some character plus .DOC." rather than having to type in the name of each file individually.

A device known as a wildcard often provides this functionality. In this problem, the only wildcard is the question mark. In the filename, the name and extension parts will contain only alphabetic characters. For example, the following names are legal:

ABC.DOG	AXC.DOG	ABCC.DOG
ABCCD.DOG	AC.DOG	AB.DOG

The pattern looks like a legal filename except that it contains 0 or more wildcards that can go anywhere an alphabetic character can go. For example, the following are legal patterns:

ABC.DOG	A?C.DOG	A??C.DOG
AB?.DOG	AB??.DOG	

The rules of matching using the wildcard are as follows.

1. Each alphabetic character in the pattern must match the same character in the filename. So pattern ABC.DOG can only be matched by filename ABC.DOG.
2. If a series of question marks in the name or extension is followed by an alphabetic character, then each wildcard must be matched by exactly one alphabetic character. So pattern A?C.DOG matches ABC.DOG and AXC.DOG but not AC.DOG or ABCC.DOG. Pattern A??C.DOG does not match ABC.DOG or AXC.DOG or AC.DOG, but does match ABCC.DOG.
3. If a series of question marks in the name or extension is not followed by an alphabetic character, then each wildcard must be matched by zero or one alphabetic character. So pattern AB??.DOG matches ABC.DOG and ABCC.DOG and AB.DOG, but not ABCCD.DOG.
4. The wildcards may appear in the name section or the extension section of the pattern or in both.

The problem.
Input: a pattern followed by a blank followed by a series of filenames separated by blanks.
Output: the filenames that match the pattern.

Example
Input: AB??.D?G AB.DAG ABC.DEF ABC.DOG ABCC.DOG ABCCD.DOG
Output: AB.DAG ABC.DOG ABCC.DOG

Problem 3: *Retirement*

A retiring religious leader has decided that his successor will be chosen from his followers in the following manner. The followers will sit around in a circle and the fatal number will be announced. Starting from the northern seat each person in the circle will count off. When someone says the fatal number, he or she leaves the circle and the next person says '1' to start the count off again. Each time the fatal number is reached, that follower leaves the circle. When there is only one person left in the circle, that person will be the new leader.

Your client is the lucky follower who gets to arrange the circle in the first place. She would like to be the next leader. Your job is to produce a program that tells her where to sit. The input to the program is two numbers, the number of followers who happen to be present and the fatal number.

Example:
Input: `10 6`
Output: `3`

Problem 4: *CODE*

A set of words is to be transmitted as messages. To save transmission time we want to represent each word by one of the letters in the word such that the words will be uniquely represented. Devise an algorithm to test if such a representation is possible for a given set of words.

Input: a set of words on a line separated by blanks.

Output: Yes or No, and with a representation if the answer is Yes.

Example:

input:

SAW WAS TODAY HANDSOM GUY

output:

Yes
S SAW
A WAS
D TODAY
M HANDSOM
U GUY

Problem 5: *Boating Navigation*

A certain river has a relatively constant current that ranges from a low of 2 mph in the Summer to as much as 6 mph in the early Spring. At one marina on the river there are several classes of boats of identical characteristics. The top speeds of the different classes range from a low of 12 mph to a high of 35 mph.

Write a C++ program to solve the following problem.

Two identical boats of the same class leave the marina at the same time with one going upstream (U-boat) and the other going downstream (D-boat). Both travel at their maximum speed. (Relative to the marina, a boat travels at its maximum speed plus or minus the speed of the current according to its direction.) Just as they leave the marina, a life preserver falls off one of the boats and begins to float downstream at the speed of the current. Several hours later, both boats reverse course. They now continue at the same maximum speed in their new direction, until one of them finds the life preserver.

Your program will get as input the speed of the river current, the speed of the boats, and the number of hours the boats travel before reversing course.

Your program should print out the following information.

1. At turnaround time
    a. Distance upstream of U-boat from the marina
    b. Distance downstream of D-boat from the marina
    c. Distance downstream of preserver from the marina
2. When the life preserver is found
    a. Distance downstream of the preserver from the marina
    b. Distance downstream of the D-boat from the marina
    c. Distance of the U-boat from the marina and whether it is upstream or downstream

Example:

Input: 3 15 4
(meaning current: 3, boat speed 15, hours to turnaround 4)

Output (values are not necessarily correct for the above input):
Turnaround: U-boat: 37. D-boat: 55. Preserver: 12
Found: Preserver: 20. D-boat: 20. U-boat: 5 downstream