

Write a complete C++ program to do the following: The main program will read in a string of three words, like "Save our ships" or "It is hot". **Make sure that there is exactly one blank between each of the parts of the string.** Then main will call several other functions to process that string. Print a few blank lines in the output and repeat the entire procedure until all strings have been read and processed.

In a loop that reads to end-of-input, the main program will read in and print a 3-word string of characters. The main program will print "Original string:" followed by the string. For each string read, the program will do the following processing:

### **breakup**

The main program will send the original string plus three empty strings to a function **breakup**. The function will break the string into three pieces, first, middle and last. These last 3 strings must be reference parameters so that the function can pass the three parts back to the main program for later use.

The **breakup** function will separate the original string into three parts at the blanks. For example, if the original string is "Wy Lee coyote", the function will separate it into "Wy", "Lee" and "coyote". (Make sure that you do not put any of the blank spaces into any of the strings.) The original string should be unchanged.

The main program will print the three parts, each on a new line, with identifying messages. For example, main could print the following:

```
Wy is first
Lee is middle
coyote is last
```

### **makealph**

The main program will then call a function **makealph** which will receive first, middle and last as parameters and will **return** a string produced from those input parameters. (This is NOT a void function.) The new string will consist of two parts of the original string joined together in alphabetical order. The two parts to be joined are these:

- the string that is closest to the beginning of the alphabet
- the string that is closest to the end of the alphabet

Remember that lowercase letters alphabetize AFTER capitals since they have larger ASCII codes, so they are considered closer to the end of the alphabet.

For example, if the original string was "Wy Lee coyote," then first is "Wy", middle is "Lee", and last is "coyote". The function will determine that the string closest to the beginning of the alphabet is "Lee", and the string closest to the end of the alphabet is "coyote". It will join these two strings as "Lee coyote" in the new string and will return this new string to main.

Main will print the string returned by **makealph**, on a new line with an identifying message.

**DATA:** Have a total of 10 or more strings. Let some strings have some words that start with capital letters, and others not. Have one string that has words that are already in alphabetical order, but make sure that the other strings use all combinations of "out of order"; that is, have the last word be alphabetically first in one; have the middle word be alphabetically first in another, etc.