American Computer Science League

2020-2021 ● Contest 2: Lex Strings ● Intermediate Division

PROBLEM: Transform a given string of all capital letters so that repeated blocks of letters are at the front, arranged such that longer blocks come first and blocks of the same length are in alphabetical order. Each string has an associated number, *m*. In the final output, all groupings of the same character must be no longer than m. Other than sorting the groups of letters that have the same frequency in the original string, no other rearranging is done.

For example, in the input line "MBAMMDXXMMMGGMMZ 3", the string contains one block of 3 letters (the M's); four blocks of 2 letters (M, X, G, and M), and 5 single letters (M, B, A, D, and Z). The 3-letter block comes first, then the 2-letter blocks (in alphabetical order), and finally, the single letters (in alphabetical order): MMMGGMMMMXXABDMZ. The number 3 requires that there is no substring of a single letter that is longer than 3 characters. In this example, MMMGGMMMXXABDMZ is output.

INPUT: Your program will receive a single line of data. Each will contain a string of no more than 100 characters, all uppercase letters followed by a space and a positive integer that will be less than the length of the string.

OUTPUT: Print each input string in the rearranged order, as described above.

SAMPLE INPUT:	**EXPECTED OUTPUT:**
MBAMMDXXMMMGGMMZ 3	MMMGGMMMXXABDMZ
MHHHHJLDDHHDDD 3	HHHDDDHHJLM
THETENNESSEEVOLUNTEERS 2	EENNSSEEHLNORSTTUV
MISSISSIPPI 3	PPSSSIIIM
BOOOKEEEPEEERR 4	EEEEOOORRBKP

American Computer Science League

2020-2021 ● Contest 2: Lex Strings ● Intermediate Division

TEST INPUT:

```
BOOOKEEEPEERBBBBUZZZOOKEEEEPEER 2
MASSACHUSETTSVSMISSISSIPPI 2
OOOOZESSPPOOOOOYYYSSSUPY 4
SHESELLSSEASHELLSANDBALLOONS 3
HHHGGRDDCFFFGGGTTTYUIKJHHH 1
```

EXPECTED OUTPUT:

```
BBEEOOZZEEOOBKKPPRRU
PPSSTTAACEHIIMMSSUV
OOOOSSSYYYPPSSEPUYZ
LLLOOSSAAABDEEEHHNNSSS
 FGHTDGCIJKRUY
```

American Computer Science League

2020-2021 ● Contest 2: Lex Strings ● Intermediate Division

问题: 转换给定的所有由大写字母组成的字符串，使重复的字符块位于前面，其中较长的字符块优先排列，长度相同的字符块按照字母顺序排列。每个字符串都有一个相关数字*m*。在最终输出结果中，所有由连续且相同的字符构成的字符块长度不能超过*m*。只需要对原字符串中出现次数相同的字符块进行排序，不进行其他重新排列。

例如，输入行中输入"MBAMMDXXMMMGGMMZ 3", 该字符串由一个包含 3 个相同字母组成的字符块（the M's）、四个包含 2 个 相 同 字母的字符块（M, X, G, 和 M）和 5 个独立字母（M, B , A, D和 Z）组成。现在将包含 3 个字母的字符块放在最前面，接着是包含 2 个字母的字符块（按照字母表顺序排列），最后是独立字母（按照字母表顺序排列），得出结果为: MMMGGMMMMXXABDMZ。数字 3 表示由单个字母构成的子字符串长度不超过 3 个字符。所以这个例子的最终输出结果 MMMGGMMMXXABDMZ 。

输入: 你将会接收到一行数据，每行包括一个不超过 100 个字符的字符串，所有大写字母后面紧跟着一个空格和一个小于字符串长度的正整数。

输出: 如上所述，按照重新排列好的顺序打印输出每个输入的字符串。

示例输入:	**预期输出:**
MBAMMDXXMMMGGMMZ 3	MMMGGMMMXXABDMZ
MHHHHJLDDHHDDD 3	HHHDDDHHJLM
THETENNESSEEVOLUNTEERS 2	EENNSSEEHLNORSTTUV
MISSISSIPPI 3	PPSSSIIIM
BOOOKEEEPEEERR 4	EEEEOOORRBKP

测试输入:

```
BOOOKEEEPEERBBBBUZZZOOKEEEEPEER 2
MASSACHUSETTSVSMISSISSIPPI 2
OOOOZESSPPOOOOOYYYSSSUPY 4
SHESELLSSEASHELLSANDBALLOONS 3
HHHGGRDDCFFFGGGTTTYUIKJHHH 1
```

测试输出：

```
BBEEOOZZEEOOBKKPPRRU
PPSSTTAACEHIIMMSSUV
OOOOSSSYYYPPSSEPUYZ
LLLOOSSAAABDEEEHHNNSSS
 FGHTDGCIJKRUY
```