

C++第一次模拟考

文本排版

给定一个单词数组和一个长度 width，首先你需要剔除掉回文单词，然后重新排版单词输出，使其成为每行恰好有 width 个字符，且左右两端对齐的文本。

你应该使用“贪心算法”来放置给定的单词(也就是说，尽可能多地往每行中放置单词)。必要时可用空格 ' ' 填充，使得每行恰好有 width 个字符。

单词与单词之间最少要有一个空格，如果需要填充多个空格，则要求尽可能**均匀分配**单词间的空格数量。如果某一行**单词间**的空格不能均匀分配，则左侧放置的空格数要多于右侧的空格数。

文本的最后一行应为左对齐，且单词之间只能有一个空格。

输出除了要将对齐的文本输出，还要对需要查询的单词进行位置的检索并输出对应单词首字母的坐标。(分别从第一行、第一列开始数)

说明:

给定的单词数组由“,”分割，单词不会包含“,”和空格字符。

每个单词的长度大于 0，小于等于 maxWidth。

输入单词数组 words 至少包含一个单词。

需要查询位置的单词一定是在对其文本中的一个单词，但是出现的次数可能不止一次，需要按照从上往下，从左向右的顺序输出位置。查询的匹配是严格匹配的，要区分大小写字母。

输入的第一行为需要处理的文本单词数组

输入的第二行为每一行的width

输入的第三行为需要查询的单词数量N

后面N行是需要查询的单词

输出:

首先输出对其的文本

然后将每一个查询单词的出现位置按照“(x,y)”的形式打印出来，一行打印一个单词的所有位置，如果一个单词有多个位置，每个位置中间按照“,”隔开

示例:

```
1  输入:
2  ["This","is","an","example","of","text","justification."]
3  16
4  1
5  is
6  输出:
7  "This    is    an"
8  "example  of text"
9  "justification. "
10 (1,9)
```

示例 2:

```
1  输入：
2  ["what", "must", "be", "acknowledgment", "level", "shall", "be", "nun"]
3  16
4  2
5  be
6  must
7  输出：
8  "what    must    be"
9  "acknowledgment  "
10 "shall be        "
11 (1,15),(3,7)
12 (1,8)
13 解释："level"和"nun"是回文，故被剔除掉。注意最后一行的格式应为 "shall be    " 而不是
    "shall        be",因为最后一行应为左对齐，而不是左右两端对齐。第二行同样为左对齐，这是因为
    这行只包含一个单词。
```