General Instructions:

- The design mark distribution for each function is given next to it. **Write the design only for those functions.**
- The floating point values in the output should be limited to two decimal places.

---

2. John has collected the prices of the laptops from different stores and prepared a list of prices. To select a laptop, he decided to sort the list in the **non-decreasing** order of price. He observed that in the list there were sequences of prices in non-decreasing order. He wanted to prepare the final sorted list of prices as follows:

   1. Find the longest sorted sequence (**non-decreasing** order) of prices from the list.
   2. Consider the 3 parts of the list as,
      - Left sub-array - the prices to the left of the longest sorted sequence (if any).
      - Middle sub-array - the longest sorted sequence.
      - Right sub-array - the prices to the right of the longest sorted sequence (if any).

      Sort the left sub-array (if any) and the right sub-array (if any) separately.
   3. Merge the three sorted sub-arrays using ***3-way merge*** into a single sorted list as described below.
      - Select the smallest price from the three sub-arrays and add it to the sorted list, until all the prices are added to the sorted list.

   Write a C program to help John to prepare the final sorted list. Your program should implement the following functions as per the given function prototypes:

   - $main()$:
     - Read the number of laptops and their prices, and store the prices in an array A.
     - Repeatedly read a character 'p', 'f', 'l', 'r', 'm', or 't' from the console and call the corresponding functions as described in Input/Output Format section, until character 't' is encountered.
   - $print\_prices(A, i, j)$: Given an array A containing the prices of n laptops, and two indices i and j such that $0 \leq i \leq j < n$, print the prices in the array A from index i to index j, separated by space. [0.5 Marks]
   - $longest\_sorted\_sequence(A, n)$: Given the prices of n laptops in the array A, find the longest sorted sequence (**non-decreasing** order) of prices. If there are more than one longest sorted sequence, consider the rightmost one in the array A.

     Note: You may use **two global variables** $start$ and end to store the starting and the ending positions of the longest sorted sequence of prices in the array A.

     [1 Mark]
   - $sort\_prices(A, n)$: Given the prices of n laptops in the array A, sort the prices in non-decreasing order.

     [1 Mark]
   - $three\_way\_merge(A, d_1, d_2, n)$: Given the prices of n laptops in the array A, and two indices d_1 and d_2 such that $0 \leq d_1 \leq d_2 <$ n. The left sub-array A[0 .. d_1], the middle sub-array A[d_1+1 .. d_2], and the right sub-array A[d_2+1 .. n-1] are sorted. Merge the three sub-arrays to form a single list sorted in **non-decreasing order**, that replaces the array A.

- In each step, let pr be the price selected for ***3-way merge*** from the three sub-arrays. If there are more than one price eligible for selection, out of them select the price from the left most sub-array as pr.
- If pr is selected from the left sub-array, print 1; if it is selected from the middle sub-array, print 2; otherwise, print 3.

[1.5 Marks]

Input/Output Format

- The first line contains an integer $n \in [1, 10^3]$ corresponding to the number of laptops.
- The second line contains n space separated floating point values corresponding to the prices (between 0 and 100, both inclusive) of the n laptops.

Each of the subsequent lines contains a character from $\{p, f, l, r, m, t\}$. For each of the characters, perform the tasks as specified below.

- Character 'p' : Print the prices of the laptops stored in the list, using $print\_prices()$ function.
- Character 'f' :
  1. Find the longest sorted sequence of prices from the list using $longest\_sorted\_sequence()$ function.
  2. Print the prices in the longest sorted sequence, using $print\_prices()$ function.
- Character 'l' :
  1. Find the longest sorted sequence of prices from the list using $longest\_sorted\_sequence()$ function.
  2. Sort the left sub-array (if any) using $sort\_prices()$ function.
- Character 'r' :
  1. Find the longest sorted sequence of prices from the list using $longest\_sorted\_sequence()$ function.
  2. Sort the right sub-array (if any) using $sort\_prices()$ function.
- Character 'm' :
  1. Find the longest sorted sequence of prices from the list using $longest\_sorted\_sequence()$ function.
  2. Sort the left sub-array (if any) and the right sub-array (if any), using $sort\_prices()$ function.
  3. Merge the three sorted sub-arrays into a single sorted list using $three\_way\_merge()$ function.
- Character 't' : Terminate the program.

Sample Input and Output

Note:

- **The floating point values in the output should be printed with two decimal places (using the format specifier `%0.2f`).**
- The sample input and output are colored ONLY for your better understanding.

- In the input, the colored elements represent the longest sorted sequence.
- In the output, the colored elements represent the sorted sub-array in that step.

Input 1

```
15
2.2 4.23 1.6523 5.3 1 2.5 3.584 3.96 6.25 7 5.063 1.45 4.9 2.2 7.451
f
l
p
r
p
m
p
t
```

Output 1

```
1.00 2.50 3.58 3.96 6.25 7.00
1.65 2.20 4.23 5.30 1.00 2.50 3.58 3.96 6.25 7.00 5.06 1.45 4.90 2.20 7.45
1.65 2.20 4.23 5.30 1.00 2.50 3.58 3.96 6.25 7.00 1.45 2.20 4.90 5.06 7.45
2 3 1 1 3 2 2 2 1 3 3 1 2 2 3
1.00 1.45 1.65 2.20 2.20 2.50 3.58 3.96 4.23 4.90 5.06 5.30 6.25 7.00 7.45
```

Input 2

```
15
2.25 1.26 7 9.13 8.25 3.25 5.68 6.36 6.87 5.9 5.36 4.1 2.3 4.6 3.36
p
f
r
p
l
p
t
```

Output 2

```
2.25 1.26 7.00 9.13 8.25 3.25 5.68 6.36 6.87 5.90 5.36 4.10 2.30 4.60 3.36
3.25 5.68 6.36 6.87
2.25 1.26 7.00 9.13 8.25 3.25 5.68 6.36 6.87 2.30 3.36 4.10 4.60 5.36 5.90
1.26 2.25 3.25 5.68 6.36 6.87 7.00 8.25 9.13 2.30 3.36 4.10 4.60 5.36 5.90
```