Problem 4 - Restricted Candies (Programming) (10 points)

Problem Description

The definition of an alternating sequence is slightly different from that of in Homework 1.

Baluteshih brings N candies to his friend, Waynetu. Those candies are lined on the table. Each candy has its own sweetness indicated by an integer. The sweetness of the candies from left to right are a_1, a_2, ..., a_N, respectively.

Baluteshih assigns Waynetu an interesting mission. He asks Waynetu to remove some candies from the table, so that the remaining candies on the table are *alternating*. Formally, we say that a sequence of candies with sweetness b_1, b_2, ..., b_k is *alternating* if $\boldsymbol{b_i \times b_{i+1} < 0}$ holds for all $1 \leq i < k$.

With the aforementioned rule, Waynetu hopes to maximize the sum of the sweetness of the candies on the table.

However, Ltf0501 comes and interrupts the mission. Because he wants to make this mission more interesting, he gives Waynetu an additional restriction, that is, **Waynetu needs to leave exactly k candies on the table**. Moreover, for each k between 1 to N, Waynetu needs to determine whether it is possible to leave exactly k candies on the table; if so, Waynetu also need to maximize the sum of the sweetness for that k value.

Please help Waynetu find the maximum possible sum of exactly k remaining candies' sweetness for each k between 1 to N.

Input

The first line contains two integers T, representing the number of test cases, and $flag$ ($flag \in \{0, 1\}$), which will be described in the output section.

Each test case includes two lines: the first line contains an integer N ($1 \leq N \leq 10^5$), and the second line contains N integers a_1, a_2, ..., a_N ($|a_i| \leq 10^9, \boldsymbol{a_i \neq 0}$).

It is guaranteed that the sum of N within the same input file does not exceed 10^5.

Test Group 0 (0 %)

- Sample Input.

Test Group 1 (20 %)

- $flag = 1$, $a_1 > 0$ and $a_N > 0$.
- $\sum N \leq 1000$.

Test Group 2 (20 %)

- $\sum N \leq 1000$.

Test Group 3 (30 %)

- $flag = 1$, $a_1 > 0$ and $a_N > 0$.

Test Group 4 (30 %)

- No additional constraints.

Output

For each test case, please print N integers in a line, separated by spaces. The i^{th} number represents the maximum possible sum of the sweetness when Waynetu leaves exactly i candies on the table. If it is impossible to leave exactly i candies on the table, just output 0 for the i^{th} number.

If $flag = 1$, then you can get Accepted when you have at least $\lfloor \frac{N}{2} \rfloor$ correct answers for each test cases.

Sample Input 1

```
1 0
5
3 -1 6 -7 4
```

Sample Output 1

```
6 5 8 2 5
```

Sample Input 2

```
2 0
3
1 2 3
4
1 -2 3 -4
```

Sample Output 2

```
3 0 0
3 1 2 -2
```

Sample Input 3

```
3 1
1
1
3
5 -1 1
4
1 2 3 4
```

Sample Output 3

```
0
5 0 0
0 0 0 0
```

Hint

1. Since each input includes several independent test cases, please carefully clear all results of the current test case before dealing with the next one.
2. In Sample Output 3, you will get Wrong Answer if $flag = 0$. The Sample Output is just a reminder for the special scoring method.
3. It is recommended to use C++ Standard Template Library (STL) data structures, such as `std::stack`, `std::queue`, `std::priority_queue`, `std::list`, `std::vector`, and `std::set`. They can reduce your coding complexity.