I. Polycarp's Practice

time limit per test: 2 seconds

memory limit per test: 256 megabytes

Polycarp is practicing his problem solving skill. He has a list of n problems with difficulties $a_1, a_2, \dots, a_n$, respectively. His plan is to practice for exactly k days. Each day he has to solve at least one problem from his list. Polycarp solves the problems in the order they are given in his list, he cannot skip any problem from his list. He has to solve all n problems in exactly k days.

Thus, each day Polycarp solves a contiguous sequence of (consecutive) problems from the start of the list. He can't skip problems or solve them multiple times. As a result, in k days he will solve all the n problems.

The *profit* of the j-th day of Polycarp's practice is the maximum among all the difficulties of problems Polycarp solves during the j-th day (i.e. if he solves problems with indices from l to r during a day, then the *profit* of the day is $\max\limits_{l \le i \le r} a_i$). The *total profit* of his practice is the sum of the *profits* over all k days of his practice.

You want to help Polycarp to get the maximum possible *total profit* over all valid ways to solve problems. Your task is to distribute all n problems between k days satisfying the conditions above in such a way, that the *total profit* is maximum.

For example, if $n = 8, k = 3$ and $a = [5, 4, 2, 6, 5, 1, 9, 2]$, one of the possible distributions with maximum *total profit* is: $[5, 4, 2], [6, 5], [1, 9, 2]$. Here the *total profit* equals $5 + 6 + 9 = 20$.

Input

The first line of the input contains two integers n and k ($1 \le k \le n \le 2000$) — the number of problems and the number of days, respectively.

The second line of the input contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 2000$) — difficulties of problems in Polycarp's list, in the order they are placed in the list (i.e. in the order Polycarp will solve them).

Output

In the first line of the output print the maximum possible *total profit*.

In the second line print exactly k positive integers $t_1, t_2, \dots, t_k$ ($t_1 + t_2 + \dots + t_k$ must equal n), where t_j means the number of problems Polycarp will solve during the j-th day in order to achieve the maximum possible *total profit* of his practice.

If there are many possible answers, you may print any of them.

Examples

input Copy

```
8 3
5 4 2 6 5 1 9 2
```

output Copy

```
20
3 2 3
```

input Copy

```
5 1
1 1 1 1 1
```

output Copy

```
1
5
```

input Copy

```
4 2
1 2000 2000 2
```

output Copy

```
4000
2 2
```

Note

The first example is described in the problem statement.

In the second example there is only one possible distribution.

In the third example the best answer is to distribute problems in the following way: $[1, 2000], [2000, 2]$. The *total profit* of this distribution is $2000 + 2000 = 4000$.

ICPC Assiut University Training - Juniors Phase 1 Sheets-2022

Public

Participant

☆

→ Group Contests

- Juniors Phase 1 Practice #5 (Bitmask, Bitset, Bits)
- Juniors Phase 1 Practice #4 (Binary search , Two pointers)
- Juniors Phase 1 Practice #3 (STL 2)
- Juniors Phase 1 Practice #2 (STL 1)
- Juniors Phase 1 Practice #1 (Prefix sum , Frequency Array)

Juniors Phase 1 Practice #1 (Prefix sum , Frequency Array)

Finished

Practice

☆

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Submit?

Language: GNU G++20 13.2 (64 bit, win

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
246039566	Feb/13/2024 00:09	Accepted
246014426	Feb/12/2024 20:02	Accepted