

HopScotch II

Time limit: 2.0s

Java: 3.0s

PyPy 2: 4.0s

PyPy 3: 4.0s

Python 2: 12.0s

Python 3: 12.0s

Memory limit: 280M

Java: 512M

Python: 1G

bruce developed a new hopscotch. In this game, a row of stones conveniently numbered 1 to N (inclusive) are set in a pool of lava. As can be expected, the stones are very hot. In fact, they can be hotter than the surface of the sun! **bruce** must jump from the start ($i = 0$) to the other side ($i > N$), but he can only jump at most K stones forward at a time. Fortunately, **bruce** has a pair of cooling boots that can cool any stone to 0 degrees. However, the boots require one unit of power per degree cooled, and power is very expensive! Can you help **bruce** find the minimum units of power required to jump to the other side of the lava pool?

bruce can only walk on stones that are 0 degrees.

bruce can only hop forwards to stone j ($i < j \leq i + K$) from stone i .

Since **bruce** starts from $i = 0$, he can hop to j ($0 < j \leq K$) on his first hop.

Input Specification

The first line will contain N and K , separated by a space.

The second line will contain all n_i ($1 \leq i \leq N$) separated by spaces.

There is a trailing newline (ASCII code 10) at the end of input.

Output Specification

Output the minimum power cost to hop from 0 to i ($i > N$) on the first line.

Output the indices of the stones **bruce** must hop on to use the minimum amount of power on the second line, separated by spaces and in ascending order.

If there are multiple ways to achieve the minimum power, output the lexicographically most sequence of the indices.

Constraints

For all subtasks:

$$0 \leq n_i < 2^{63}$$

$$0 \leq \sum_{i=1}^N n_i < 2^{63}$$

$$1 \leq N \leq 2^{23}$$

$$1 \leq K \leq N$$

Subtask 1 [1/15]

$$K = N$$

Subtask 2 [1/15]

$$K = 1$$

Subtask 3 [3/15]

$$K = \sqrt{N}$$

Subtask 4 [10/15]

No additional constraints.

Sample Input 1

```
16 4
4 5 3 12 2 6 3 6 5 5 16 1 10 9 13 12
```

Sample Output 1

```
20
3 5 9 12 14
```

Sample Input 2

```
16 2
4 13 6 6 4 1 7 1 0 15 3 0 8 11 5 8
```

Sample Output 2

```
32
1 3 5 6 8 9 11 12 13 15
```