Problem 1. Maximum sum with swaps

Input file:	input.txt
Output file:	output.txt
Time limit:	6 seconds
Memory limit:	512 megabytes

Given an array of integers A_i of length N, you are to find an interval within the array, which has maximum sum of the elements in it. However, before choosing the interval, you are allowed to perform no more than K swaps. Each swap makes two elements of the array exchange places.

An interval within the array is an arbitrary set of its **consecutive** elements.

Input

The first line of the input file contains two integers: N — the number of elements in the array ($1 \le N \le 100\,000$), K — the maximum possible number of swaps ($0 \le K \le 10$). The second line lists the elements of the array A_i, one by one. All A_i are integers, no larger than 10^9 by absolute value. It is guaranteed that there is at least one positive number among A_i.

Output

The first line of the output file must contain two integers: S — the resulting sum of elements in the interval and M — the number of swaps performed ($0 \le M \le K$).

The following M lines must describe all swaps in order of their execution. For every j-th swap print two integers u_j and v_j, denoting two positions in the array, whose values are to be swapped ($1 \le u_j \ne v_j \le N$). The positions in the array are numbered consecutively starting from one.

The last line must contain the interval where the sum must be calculated, described as two integers: L being the index of the leftmost position in the array belonging to the interval, and R being the index of the rightmost position belonging to the interval ($1 \le L \le R \le N$).

Examples

input.txt	output.txt
3 2 1 2 3	6 0 1 3
3 3 1 -2 3	4 2 1 2 2 3 2 3
3 0 1 -2 3	3 0 3 3