For a sequence of **N** integers – a0, a1, …, аn-1 we define the **k**-area of a number to be to be the sum of the number with index **i**, the **k** numbers to its left and the **k** numbers to its right. Write a program, which for a given sequence of numbers and number **k** outputs the index of the number with the largest k-area. If there is more than one largest area, output the smallest index.

**Note: if before or after a given index there are less than k numbers, we only take into account their sum when calculating the k-area (i.e. we can imagine that the missing numbers are all 0).**

**Input**

The first row of the file karea.in contains two positive integers **N** and **k** – the length of the sequence and the size of the area respectively.

**N** integers follow – the numbers of the sequence.

**Output**

In the output file karea.out print the position of the number with the largest **k**-area. In case there is more than one such are, print the smallest index. (Indexing begins at 0).

**Constraints**

3 ≤ *N* ≤ 106

3 ≤ *k* ≤ 5\*105

-1000 ≤ *ai* ≤ 1000

**Time limit: 0.5 sec**

**Memory limit: 256 MB**

**Example test**

|  |  |
| --- | --- |
| **Input (karea.in)** | **Output (karea.out)** |
| 5 1  -10 9 -10 -5 -9 | 0 |
| 6 4  -6 10 -9 -5 -6 3 | 5 |

**Clarifications**

Example 1: The values of the 1-areas in each position are as follows:

0: -10 + 9 = -1

1: -10 + 9 + (-10) = -11

2: 9 + (-10) + (-5) = -6

3: -10 + (-5) + (-9) = -24

4: -5 + (-9) = -14

The largest value is -1 at index 0.