Dominoes in a Line

Statement

Fluffy the Hamster has N dominoes. No two dominoes have the same height.

Fluffy wants to arrange these dominoes in a line with no gaps in between. Each domino's front face should be facing him. How many ways can Fluffy arrange them, such that exactly K dominoes have their front face completely covered by the one in front of it?

Input

The first line consists of two integers `N K`.

The second line consists of N space-separated integers, the heights of the dominoes H_i.

Constraints

- $1 \leq N \leq 10$
- $0 \leq K \leq 10$
- $1 \leq H_i \leq 10^3$
- For distinct dominoes i and j, $H_i \neq H_j$.

Output

Print the number of ways to arrange the dominoes, such that exactly K front faces are completely covered.

Examples

Sample Input	Expected Output
3 1 1 4 9	4
5 2 4 7 12 34 13	66
3 3 2 4 6	0

In the first input, there are 4 ways to have exactly 1 front face covered up:

$$(1, 9, 4), (4, 1, 9), (4, 9, 1), (9, 1, 4)$$

In the third case, any arrangement of 3 dominoes will result in at most 2 faces covered. There are **0** possible arrangement satisfying the constraints.

Notes

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided.

Skeleton File

You are given the skeleton file `Dominoes.java`. You should see the following contents when you open the file:

```
/**
 * Name          :
 * Matric. No    :
*/

import java.util.*;
```

```java
public class Dominoes {
    private void run() {
        // implement your "main" method here
    }

    public static void main(String args[]) {
        Dominoes runner = new Dominoes();
        runner.run();
    }
}
```