## Problem 1 - Powers

Numbers have powers! They can transform themselves. One transformation is done by replacing:

* each 0 - with the absolute difference of its neighboring numbers
* all other even numbers - with the maximum of its neighboring numbers
* each 1 - with the sum of its neighboring numbers
* all other odd numbers - with the minimum of its neighboring numbers

The leftmost and rightmost numbers **are** neighbors.

A K-sum of a sequence is the sum of the numbers after **K** transformations of the sequence. Your task is to find the K-sum of a given sequence.

### Input

The input data is given as a parameter – an array of strings.

On the first input line there will be the numbers **N** and **K** separated by a space. On the second input line are **N** numbers – the sequence.

### Output

The output should be printed on the console.

Output the K-sum of the given sequence.

### Sample solution code (in JavaScript)

function solve(params) {

var nk = params[0].split(' ').map(Number),

s = params[1].split(' ').map(Number),

result;

// Your solution here

console.log(result);

}

### Constraints

* **3 <= N <= 100**
* **0 <= K <= 20**
* Initially, each number in the sequence is a single digit non-negative integer
* Allowed working time for your program: **0.3 seconds**.
* Allowed memory: **16 MB**.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Explanation** |
| **5 1**  **9 0 2 4 1** | **26** | **9 0 2 4 1**  **becomes**  **0 7 4 2 13** |
| **10 3**  **1 9 1 9 1 9 1 9 1 9** | **365** | **1 9 1 9 1 9 1 9 1 9**  **becomes**  **18 1 18 1 18 1 18 1 18 1**  **and then**  **1 36 1 36 1 36 1 36 1 36**  **and then**  **72 1 72 1 72 1 72 1 72 1** |
| **10 10**  **0 1 2 3 4 5 6 7 8 9** | **42** |  |