Greedy Florist ☆

146/563 challenges solved

Rank: **3836** | Points: **3492.67**

Problem Solving

Problem Submissions Leaderboard Editorial Topics

A group of friends want to buy a bouquet of flowers. The florist wants to maximize his number of new customers and the money he makes. To do this, he decides he'll multiply the price of each flower by the number of that customer's previously purchased flowers plus 1. The first flower will be original price, $(0+1) \times \textbf{original price}$, the next will be $(1+1) \times \textbf{original price}$ and so on.

Given the size of the group of friends, the number of flowers they want to purchase and the original prices of the flowers, determine the minimum cost to purchase all of the flowers.

For example, if there are $k=3$ friends that want to buy $n=4$ flowers that cost $c=[1,2,3,4]$ each will buy one of the flowers priced $[2,3,4]$ at the original price. Having each purchased $x=1$ flower, the first flower in the list, $c[0]$, will now cost $(current\ purchase + previous\ purchases) \times c[0] = (1+1) \times 1 = 2$. The total cost will be $2+3+4+2=11$.

Function Description

Complete the getMinimumCost function in the editor below. It should return the minimum cost to purchase all of the flowers.

getMinimumCost has the following parameter(s):

- c: an array of integers representing the original price of each flower
- k: an integer, the number of friends

Input Format

The first line contains two space-separated integers n and k, the number of flowers and the number of friends.
The second line contains n space-separated positive integers $c[i]$, the original price of each flower.

Constraints

- $1 \le n, k \le 100$
- $1 \le c[i] \le 10^6$
- $answer < 2^{31}$
- $0 \le i < n$

Output Format

Print the minimum cost to buy all n flowers.

Sample Input 0

```
3 3
2 5 6
```

Sample Output 0

```
13
```

Explanation 0

There are $n = 3$ flowers with costs $c = [2, 5, 6]$ and $k = 3$ people in the group. If each person buys one flower, the total cost of prices paid is $2 + 5 + 6 = 13$ dollars. Thus, we print 13 as our answer.

Sample Input 1

```
3 2
2 5 6
```

Sample Output 1

```
15
```

Explanation 1

There are $n = 3$ flowers with costs $c = [2, 5, 6]$ and $k = 2$ people in the group. We can minimize the total purchase cost like so:

1. The first person purchases 2 flowers in order of decreasing price; this means they buy the more expensive flower ($c_1 = 5$) first at price $p_1 = (0 + 1) \times 5 = 5$ dollars and the less expensive flower ($c_0 = 2$) second at price $p_0 = (1 + 1) \times 2 = 4$ dollars.
2. The second person buys the most expensive flower at price $p_2 = (0 + 1) \times 6 = 6$ dollars.

We then print the sum of these purchases, which is $5 + 4 + 6 = 15$, as our answer.

Sample Input 2

```
5 3
1 3 5 7 9
```

Sample Output 2

```
29
```

Explanation 2

The friends buy flowers for 9, 7 and 3, 5 and 1 for a cost of $9 + 7 + 3 * (1 + 1) + 5 + 1 * (1 + 1) = 29$.

Current Buffer (saved locally, editable)

Java 8

```
import java.io.*;
import java.math.*;
```

```java
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

public class Solution {

    // Complete the getMinimumCost function below.
    static int getMinimumCost(int k, int[] c) {


    }

    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        String[] nk = scanner.nextLine().split(" ");

        int n = Integer.parseInt(nk[0]);

        int k = Integer.parseInt(nk[1]);

        int[] c = new int[n];

        String[] cItems = scanner.nextLine().split(" ");
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

        for (int i = 0; i < n; i++) {
            int cItem = Integer.parseInt(cItems[i]);
            c[i] = cItem;
        }

        int minimumCost = getMinimumCost(k, c);

        bufferedWriter.write(String.valueOf(minimumCost));
        bufferedWriter.newLine();

        bufferedWriter.close();

        scanner.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File □ Test against custom input Run Code Submit Code

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature