



# Greedy Florist

Problem

Submissions

Leaderboard

Discussions

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,  $(0 + 1) \times \text{original price}$ , the next will be  $(1 + 1) \times \text{original price}$  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  $(\text{current purchase} + \text{previous purchases}) \times c[0] = (1 + 1) \times 1 = 2$ . 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

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

## 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  $9 + 7 + 3 * (1 + 1) + 5 + 1 * (1 + 1) = 29$ .

f t in

Contest ends in 9 hours

Submissions: [249](#)



Max Score: 25




Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

[More](#)

Current Buffer (saved locally, editable)  

C++14   

```

1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 vector<string> split_string(string);
6
7 // Complete the getMinimumCost function below.
8 int getMinimumCost(int k, vector<int> c) {
9
10
11 }
12
13 int main()
14 {
15
16
17 vector<string> split_string(string input_string) {
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

```

 [Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code