

[Practice](#)[Compete](#)[Jobs](#)[Rank](#)[Leaderboard](#)

gadhiya

[Dashboard](#) > [Algorithms](#) > [Dynamic Programming](#) > Turn Off the Lights

Badge Progress



Points: 4727.88 Rank: 491

Turn Off the Lights



by coolbun

Problem

[Submissions](#)[Leaderboard](#)[Discussions](#)[Editorial](#)

There are n bulbs in a straight line, numbered from 0 to $n - 1$. Each bulb i has a button associated with it, and there is a cost, c_i , for pressing this button. When some button i is pressed, all the bulbs at a distance $\leq k$ from bulb i will be toggled(off->on, on->off).

Given n , k , and the costs for each button, find and print the minimum cost of turning off all n bulbs if they're all on initially.

Input Format

The first line contains two space-separated integers describing the respective values of n and k .

The second line contains n space-separated integers describing the respective costs of each bulb (i.e., c_0, c_1, \dots, c_{n-1}).

Constraints

- $3 \leq n \leq 10^4$
- $0 \leq k \leq 1000$
- $0 \leq c_i \leq 10^9$

Output Format

Print a long integer denoting the minimum cost of turning off all n bulbs.

Sample Input

```
3 1
1 1 1
```

Sample Output

```
1
```

Explanation

If we press the middle switch, the middle bulb and the $k = 1$ closest adjacent bulbs (i.e., the first and third) will turn off. Because all bulbs will be off in one button press, this cost is minimal. Thus, we print **1** as our answer.

[f](#) [t](#) [in](#)


Submissions: 106

Max Score: 70

Difficulty: Hard

Rate This Challenge:

[More](#)

Current Buffer (saved locally, editable)  

Java 8



```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) throws IOException {
7
8         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
9         String[] num = br.readLine().split("\\s");
10
11         int n = Integer.parseInt(num[0]);
12         int k = Integer.parseInt(num[1]);
13
14         num = br.readLine().split("\\s");
15
16         long[] arr = new long[n];
17
18         for(int i = 0 ; i < n ; i++){
19             arr[i] = Long.parseLong(num[i]);
20         }
21
22         long output = Integer.MAX_VALUE;
23
24         for(int i = 1 ; i <= k+1 ; i++){
25
26             int index = i;
27             long cost = 0;
28
29             while(index < n ){
30                 cost+= arr[index];
31                 index+=(2*k + 1);
32             }
33
34             index = index - 2*k - 1 ;
35
36             if(index+k >= n){
37                 output = Math.min(output,cost);
38             }
39
40         }
41
42         System.out.println(output);
43     }
44 }
45
46 }
```

Line: 1 Col: 1

 Upload Code as File☐ Test against custom input

Run Code

Submit Code

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)