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., $c_0, c_1, \ldots, c_{n-1}$).

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.

Current Buffer (saved locally, editable) Java 8

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) throws  IOException {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String[] num = br.readLine().split("\\s");

        int n = Integer.parseInt(num[0]);
        int k = Integer.parseInt(num[1]);

        num = br.readLine().split("\\s");

        long[] arr = new long[n];

        for(int i = 0 ; i < n ; i++){
            arr[i] = Long.parseLong(num[i]);
        }

        long output = Integer.MAX_VALUE;

        for(int i = 1 ; i <= k+1 ; i++){

            int index = i;
            long cost = 0;

            while(index < n ){
                cost+= arr[index];
                index+=(2*k + 1);
            }


            index = index - 2*k - 1 ;

            if(index+k >= n){
                output = Math.min(output,cost);
            }

        }


        System.out.println(output);
    }
}
```

Line: 1 Col: 1

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