**Maximize sum after K negations**

Show Topic Tags   

Given an array of size n and a number k. We must modify array K number of times. Here modify array means in each operation we can replace any array element arr[i] by -arr[i]. We need to perform this operation in such a way that after K operations, sum of array must be maximum?

Examples:

Input : arr[] = {-2, 0, 5, -1, 2}

K = 4

Output: 10

// Replace (-2) by -(-2), array becomes {2, 0, 5, -1, 2}

// Replace (-1) by -(-1), array becomes {2, 0, 5, 1, 2}

// Replace (0) by -(0), array becomes {2, 0, 5, 1, 2}

// Replace (0) by -(0), array becomes {2, 0, 5, 1, 2}

Input : arr[] = {9, 8, 8, 5}

K = 3

Output: 20

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case consist of three lines . The first line of each test case contains an integer N.The second line of each test case contains N space separated integers denoting elements of the array. Third line contains value of k.  
  
**Output:**  
For each test case in a new line print maximum possible sum.  
  
**Constraints:**  
1 ≤ T ≤ 500  
1 ≤ N ≤ 1000  
  
**Example:**  
**Input:**  
2  
5  
1 2 -3 4 5  
1  
10  
5 -2 5 -4 5 -12 5 5 5 20  
5

**Output:**  
15  
68

\*\*For More Examples Use Expected Output\*\*

<http://practice.geeksforgeeks.org/problems/maximize-sum-after-k-negations/0>

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class GFG {

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

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

String[] input = br.readLine().trim().split(" ");

int[] arr = new int[n];

int negativos =0;

for(int i =0; i<n; i++) {

arr[i] = Integer.parseInt(input[i]);

if(arr[i] < 0) {

negativos++;

}

}

int k = Integer.parseInt(br.readLine());

Arrays.sort(arr);

int i =0;

if(k >= negativos) {

while(i < arr.length && i<k && arr[i] < 0 ) {

arr[i]\*=-1;

i++;

}

int sobra = k - negativos;

if(sobra %2 !=0) {

arr[i] \*= -1;

}

}

else if(k < negativos) {

i=0;

while(i < arr.length && i<k && arr[i] < 0 ) {

arr[i]\*=-1;

i++;

}

}

int sum =0;

for( i =0; i<arr.length; i++) {

sum += arr[i];

}

System.out.println(sum);

}

}

}