1. Sorted Sums

For a sequence of integers $[a_1, a_2, \ldots a_n]$, define the function $f(i)$ as follows:

- Take the first i elements of a $(a_1, a_2, \ldots a_i)$ and sort them in non-descending order. Call this new sequence s_i.
- Let $f(i) = 1*s_1 + 2*s_2 + \ldots i*s_i$

Given a sequence of n integers, sort them in non-descending order then compute $f(1)+f(2)+f(3)+\ldots f(n)$. As the result may be very large, return it modulo $(10^9 + 7)$.

Example
$n = 4$
$a = [4, 3, 2, 1]$

$s_1 = [4]$, $f(1) = 1*4 = 4$
$s_2 = [3,4]$, $f(2) = 1*3+2*4 = 11$
$s_3 = [2,3,4]$, $f(3) = 1*2+2*3+3*4 = 20$
$s_4 = [1,2,3,4]$, $f(4) = 1*1+2*2+3*3+4*4 = 30$
$f(1)+f(2)+..f(n) = 4+11+20+30 = 65$ and 65 modulo $(10^9 + 7) = 65$

Function Description
Complete the function *sortedSum* in the editor below. The function should return the value of $f(1)+f(2)+....f(n)$ modulo $(10^9 + 7)$.

sortedSum has the following parameter(s):
a: a sequence of integers

Constraints

- $1 \le n \le 10^5$
- $1 \le a[i] \le 10^6$

▼ Input Format For Custom Testing

The first line contains an integer, n, denoting the number of elements in a.

Each line i of the n subsequent lines (where $1 \le i \le n$) contains an integer describing $a[i]$.

▼ Sample Case 0

Sample Input For Custom Testing

```
3
9
5
8
```

Sample Output

```
80
```

Explanation
$n = 3$
$a = [9, 5, 8]$

- $s_1 = [9]$; $f(1) = 1*9 = 9$
- $s_2 = [5,9]$; $f(2) = 1*5 + 2*9 = 23$
- $s_3 = [5,8,9]$; $f(3) = 1*5 + 2*8 + 3*9 = 48$
- $f(1)+f(2)+....f(n) = f(1)+f(2)+f(3) = 9 + 23 + 48 = 80$

80 modulo $(10^9 + 7) = 80$

▼ Sample Case 1

Sample Input For Custom Testing

```
2
5
9
```

Sample Output

```
28
```

Explanation

n = 2
a = [5, 9]

- *f*(1) = 1*5 = 5
- *f*(2) = 1*5+2*9 = 23
- *f*(1)+*f*(2)+....*f*(*n*) = *f*(1)+*f*(2) = 5 + 23 = 28

28 modulo $(10^9 + 7) = 28$