Selection Sort

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Every student enrolled in the algorithms course is required to submit an assignment this week. The task is to implement an $O(n^2)$-time algorithm to sort n given integers in non-decreasing order. Alice has already completed her assignment, and her implementation is shown below.

```
int alice_sort(int *s, int n){
  for(int i = 0; i < n; ++i){
    for(int j = i + 1; j < n; ++j){
      if(s[i] > s[j]){
        int swap = s[i];
        s[i] = s[j];
        s[j] = swap;
      }
    }
  }
  return 0;
}
```

While you have access to Alice's code, you prefer not to simply copy it. Instead, you want to use Alice's sorting function as a building block for your own solution. There are two ways as listed below you can utilize her function, but each of them can be applied at most **once**. The order in which these two operations are invoked can be arbitrary.

- Prefix sort: choose a length $i \in \{1, 2, \ldots, n\}$ and call `alicesort(`s, i`)`. This sorts the first i elements in the array s.
- Suffix sort: choose a length $i \in \{1, 2, \ldots, n\}$ and call `alicesort(`$s + n - i, i$`)`. This sorts the last i elements in the array s.

Due to the time complexity of the sorting algorithm, the cost of performing either a prefix or suffix sort is i^2, where i is the length of the chosen subarray. Your goal is to determine the minimum cost to sort the input array s of n integers in non-decreasing order using Alice's function, following the rules mentioned above.

For example, Let $s = [3, 2, 5, 5, 4, 1]$. We can first perform a suffix sort of length 4, and the array becomes $[3, 2, 1, 4, 5, 5]$. Then, we perform a prefix sort of length 3, and the array becomes $[1, 2, 3, 4, 5, 5]$, which is a sorted array. The cost is $4^2 + 3^2 = 25$. Here is another example, let $s = [4, 3, 2, 1]$. We can complete the sorting by performing only a prefix sort of length 4, and the cost is $4^2 = 16$.

Input

The first line contains exactly one integer n which indicates the number of integers in the array s. The second line contains the n integers in $s = [s_0, s_1, \ldots, s_{n-1}]$.

- $1 \leq n \leq 10^6$
- For all i $(0 \leq i < n)$, $0 \leq s_i < 2^{31} - 1$.

Output

Output an integer on a line, indicating the minimum cost to sort the input array s of n integers in non-decreasing order using Alice's function, following the rules mentioned above.

Examples

standard input	standard output
6 3 2 5 5 4 1	25
4 4 3 2 1	16