Sorting is not easy

An N-element permutation is an N-element sequence of distinct numbers from the set {1, 2 ... n}. For example the sequence 2, 1, 4, 5, 3 is a 5-element permutation. P is an N-element permutation. Your task is to sort P in ascending order. But because it is very simple, I have a new rule for you. You have two sequences P and Q. P is an N-element permutation and Q is initially empty and formed by sorting P (i.e. finally Q = 1, 2, 3 ... N). You have to implement N steps to sort P. In the i-th step, P has N-i+1 remaining elements, Q has i-1 elements and you have to choose some x-th element (from the N-i+1 available elements) of P and put it to the left or to the right of Q. The cost of this step is equal to $x * i$. The total cost is the sum of costs of individual steps. After N steps, Q must be an ascending sequence. Your task is to minimize the total cost.

Input

The first line of the input file is T ($T \leq 10$), the number of test cases. Then descriptions of T test cases follow. The description of each test case consists of two lines. The first line contains a single integer N ($1 \leq N \leq 1000$). The second line contains N distinct integers from the set {1, 2 ... N}, the N-element permutation P.

Output

For each test case your program should write one line, containing a single integer - the minimum total cost of sorting.

Example

N = 4
P = {4,1,3,2}
Step 1, Choose 3-rd, P={4,1,2}, Q={3} , Cost=3
Step 2, Choose 1-st, P={1,2}, Q={3,4} , Cost=2
Step 3, Choose 2-nd, P={1}, Q={2,3,4} , Cost=6
Step 4, Choose 1-st, P={}, Q={1,2,3,4}, Cost=4
The total cost is 15.
Another way to sort:
Step 1, Choose 4-th, P={4,1,3}, Q={2} , Cost=4
Step 2, Choose 2-nd, P={4,3}, Q={1,2} , Cost=4
Step 3, Choose 2-nd, P={4}, Q={1,2,3} , Cost=6
Step 4, Choose 1-st, P={}, Q={1,2,3,4}, Cost=4
The total cost is 18.

Input:
1
4
4 1 3 2

Output:
15