Problem H
Merge Sort

POTYCZKI ALGORYTMICZNE

The 3rd Universal Cup, Stage 40: Potyczki. Limits: 1024 MB, 3 s. *21.06.2025*

Merge sort is one of the most well-known sorting algorithms. In this task, we use an implementation described by the following pseudocode:

Algorithm: Implementation of the merge sort algorithm.

```
1:  function MergeSort(p, n)          ▷ Returns a sorted version of the list p = [p[0], p[1], ..., p[n − 1]].
2:      if n = 1 then
3:          return p
4:      left ← MergeSort([p[0], ..., p[⌈n/2⌉ − 1]], ⌈n/2⌉)
5:      right ← MergeSort([p[⌈n/2⌉], ..., p[n − 1]], ⌊n/2⌋)
6:      (i, j) ← (0, 0)
7:      result ← []                                   ▷ [] denotes an empty list.
8:      while i < |left| and j < |right| do           ▷ |left|, |right| are lengths of lists left, right.
9:          if left[i] < right[j] then
10:             Append the value left[i] to the end of list result.
11:             i ← i + 1
12:         else
13:             Append the value right[j] to the end of list result.
14:             j ← j + 1
15:     Append the values left[i], ..., left[|left| − 1] to the end of list result.
16:     Append the values right[j], ..., right[|right| − 1] to the end of list result.
17:     return result
```

You are given numbers n, a, and b. Your task is to count such permutations of numbers from 1 to n that sorting the permutation using the MergeSort procedure performs a comparison of a and b at least once in line 9. Output the remainder of the division of the number of such permutations by $1\,000\,000\,007$ ($10^9 + 7$).

Note that a permutation should be counted if numbers a and b are compared in any recursive call of the MergeSort procedure, not necessarily in the shallowest call. Also note that we allow comparison of numbers a and b in any order. In other words, checking only one condition $a < b$ or $b < a$ is enough to count the permutation.

Additionally, you need to solve multiple test cases.

Input

The first line of the input contains an integer t ($1 \le t \le 10\,000$), representing the number of test cases.

The next t lines contain descriptions of consecutive test cases. The i-th line contains three integers n, a, and b ($2 \le n \le 1\,000\,000; 1 \le a, b \le n; a \ne b$), as described in the task.

Output

The output should consist of t lines. The i-th line should contain a single integer – the remainder of the division by $10^9 + 7$ of the number of sought permutations in the i-th test case.

Example

For the input data:	the correct result is:
3 4 2 3 5 4 1 3 1 3	24 52 4

Explanation: In the first test case, the numbers 2 and 3 would be compared during the sorting of all permutations of length 4.

In the third test case, we will compare the numbers 1 and 3 during the sorting of permutations $[1, 2, 3]$, $[1, 3, 2]$, $[2, 1, 3]$ and $[3, 1, 2]$.