Given a permutation, produce its [inverse permutation](keyword://inverse-permutation" \t "_blank).

**Example**

For permutation = [1, 3, 4, 2], the output should be  
inversePermutation(permutation) = [1, 4, 2, 3].

Input/Output

* **[execution time limit] 0.5 seconds (cpp)**
* **[input] array.integer permutation**

A non-empty array representing a permutation of integers from 1 to some n.

*Guaranteed constraints:*  
3 ≤ permutation.length ≤ 10.

* **[output] array.integer**
  + The inverse permutation of permutation.

**[C++] Syntax Tips**

// Prints help message to the console

// Returns a string

std::string **helloWorld**(std::string name) {

std::cout << "This prints to the console when you Run Tests" << std::endl;

**return** "Hello, " + name;

}

<https://app.codesignal.com/challenge/A674cE5ibcAEabYZn>

std::vector<int> inversePermutation(std::vector<int> permutation) {

vector<int> ans;

int size = permutation.size();

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

ans.push\_back(0);

}

for(int i = 1; i <= size; i++) {

ans[permutation[i - 1] - 1] = i;

}

return ans;

}