If some element of the given array is prime, and its index is prime too, then this is a *very prime* element.  
Given array, return the number of *very prime*elements in it. The first index of the array is 1.  
  
**Examples**  
  
For array = [3, 3, 7, 2, 13, 4, 2, 4, 4] the answer should be 4 (array[2] = 3, array[3] = 7,array[5] = 13 and array[7] = 2).  
  
For array = [3, 3, 2, 2, 1] the answer should be2.

* **[input] array.integer array**
  + 1 ≤ arrayi ≤ 100. *(1 ≤ array.size ≤ 100)*
* **[output] integer**
  + The number of very prime elements.

<https://codefights.com/challenge/WTxLgNbKCFTsNdHT8>

--ACEPTADO—

#include <conio.h>

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <vector>

using namespace std;

int primes(std::vector<int> array) {

struct Helper{

bool esPrimo(int n){

if(n < 2)return false;

if(n == 2) return true;

if(n %2 == 0)return false;

int sqr = (int)sqrt(n);

for(int i =3; i<=sqr; i+=2) {

if(n%i==0) return false;

}

return true;

}

};

Helper h;

int cont = 0;

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

if(h.esPrimo(i+1) && h.esPrimo(array[i])){

cont ++;

}

}

return cont;

}

int main(){

int arr[] = {3, 3, 7, 2, 13, 4, 2, 4, 4};

int n = sizeof(arr)/sizeof(int);

std::vector<int> v;

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

v.push\_back(arr[i]);

}

int res = primes(v);

printf("%d ", res);

getch();

return 0;

}