
Reorganize a vector**X12559_en**

Write a function `reorganize_vector(v, a, b)` that receives a vector $v = (v_0, \dots, v_{n-1})$ of integers and two integers $a \leq b$ as parameters, and returns a vector w that contains the same elements as v but reorganized as follows: on the first positions of w there are the elements v_i such that $v_i < a$ (and preserving their original respective order in v between them), next there are the elements v_i such that $a \leq v_i \leq b$ (again preserving the original order in v), and finally the elements v_i satisfying $b < v_i$ (again preserving the original order in v).

For example, for $v = [3, 6, 2, 7, 5, 1, 4, 3, 1, 7, 2, 5, 6, 4, 3]$, $a = 3$, $b = 5$, the subvector of elements strictly smaller than a is $[2, 1, 1, 2]$, the subvector of elements between a and b is $[3, 5, 4, 3, 5, 4, 3]$, and the subvector of elements strictly greater than b is $[6, 7, 7, 6]$. Thus, the function must return $w = [2, 1, 1, 2, 3, 5, 4, 3, 5, 4, 3, 6, 7, 7, 6]$.

Your submission must use the following C++ code, modifying only the parts that are explicitly indicated.

```
#include <iostream>
#include <vector>

using namespace std;

// Returns a vector of size n > 0 with n integer values read from cin.
vector<int> read_vector(int n) {
    vector<int> v(n);
    for (int i = 0; i < n; ++i)
        cin >> v[i];
    return v;
}

// Prints a vector of size n > 0 in the cout.
void print_vector(const vector<int>& v) {
    for (int i = 0 ; i < int(v.size()) ; ++i) {
        if (i>0) cout<<" ";
        cout<<v[i];
    }
    cout<<endl;
}

// add functions or procedures here if you need them

// see the statement of the problem
vector<int> reorganize_vector(const vector<int>& v,int a,int b)
{
    // Write your implementation here.
}

int main() {
    int n;
```

```

while (cin >> n) {
    vector<int> v = read_vector(n);
    int a, b;
    cin >> a >> b;
    print_vector(reorganize_vector(v, a, b));
}
}

```

CASTELLANO

Escribe una función `reorganize_vector(v, a, b)` que recibe un vector $v = (v_0, \dots, v_{n-1})$ de enteros y dos enteros $a \leq b$ como parámetros, y devuelve un vector w que contiene los mismos elementos que v pero reorganizados como sigue: las primeras posiciones de w contienen los elementos v_i tales que $v_i < a$ (preservando su orden original respectivo en v), a continuación vienen los elementos v_i tales que $a \leq v_i \leq b$ (nuevamente preservando su orden original en v) y finalmente están los elementos v_i tales que $b < v_i$ (nuevamente preservando su orden original en v).

Por ejemplo, para $v = [3, 6, 2, 7, 5, 1, 4, 3, 1, 7, 2, 5, 6, 4, 3]$, $a = 3$ y $b = 5$, el subvector de elementos estrictamente menores que a es $[2, 1, 1, 2]$, el subvector de elementos entre a y b es $[3, 5, 4, 3, 5, 4, 3]$, y el subvector de elementos estrictamente mayores que b es $[6, 7, 7, 6]$. Por lo tanto, la función debe devolver $w = [2, 1, 1, 2, 3, 5, 4, 3, 5, 4, 3, 6, 7, 7, 6]$.

Tu envío (*submission*) debe utilizar obligatoriamente el código en C++ dado más arriba, modificando sólo las partes explícitamente indicadas.

CATALÀ

Escriu una funció `reorganize_vector(v, a, b)` que rep un vector $v = (v_0, \dots, v_{n-1})$ d'enters i dos enters $a \leq b$ com paràmetres, i retorna un vector w que conté els mateixos elements que v però reorganitzats com segueix: les primeres posicions de w contenen els elements v_i tals que $v_i < a$ (preservant el seu ordre original respectiu en v), a continuació vénen els elements v_i tals que $a \leq v_i \leq b$ (novament preservant el seu ordre original en v) i finalment hi ha els elements v_i tals que $b < v_i$ (novament preservant el seu ordre original en v).

Per exemple, per $v = [3, 6, 2, 7, 5, 1, 4, 3, 1, 7, 2, 5, 6, 4, 3]$, $a = 3$ i $b = 5$, el subvector d'elements estrictament menors que a és $[2, 1, 1, 2]$, el subvector d'elements entre a i b és $[3, 5, 4, 3, 5, 4, 3]$, i el subvector d'elements estrictament més grans que b és $[6, 7, 7, 6]$. Per tant, la funció ha de retornar $w = [2, 1, 1, 2, 3, 5, 4, 3, 5, 4, 3, 6, 7, 7, 6]$.

El teu enviament (*submission*) ha d'utilitzar obligatòriament el codi en C++ donat més amunt, modificant només les parts explícitament indicades.

Exam score: 2.5 **Automatic part:** 100%

Input

The input is a sequence of cases. Each case is described with three lines. The first line has an integer $n > 0$. The second line has n integers, that is the contents of a vector v of size n , and the third line has two more integers a and b holding $a \leq b$.

CASTELLANO

La entrada es una secuencia de casos. Cada caso se describe en tres líneas. La primera línea contiene un entero $n > 0$. La segunda línea contiene n enteros, el contenido del vector v de tamaño n , y la tercera línea contiene dos enteros más, a y b , tales que $a \leq b$.

=====

CATALÀ

=====

L'entrada és una seqüència de casos. Cada cas es descriu en tres línies. La primera línia conté un enter $n > 0$. La segona línia conté n enters, el contingut del vector v de mida n , i la tercera línia conté dos enters més, a i b , tals que $a \leq b$.

Output

For each case, the program outputs the contents of the resulting vector from a call `reorganize_vector(v, a, b)` followed by a line break. Two consecutive elements are separated by a white space.

=====

CASTELLANO

=====

Para cada caso, el programa imprime el contenido del vector retornado por la llamada `reorganize_vector(a, b)`, seguidos por un salto de línea. Cualquier par de elementos consecutivos se imprimen separados por un espacio en blanco.

=====

CATALÀ

=====

Per a cada cas, el programa imprimeix el contingut del vector retornat per la crida `reorganize_vector(v, a, b)`, seguits per un salt de línia. Qualsevol parell d'elements consecutius s'imprimeixen separats per un espai en blanc.

Sample input 1

```
15
3 6 2 7 5 1 4 3 1 7 2 5 6 4 3
3 5
10
4 -3 5 7 1 4 8 6 5 -7
3 5
10
4 -3 5 7 1 4 8 6 5 -7
-10 -9
10
4 -3 5 7 1 4 8 6 5 -7
2 3
10
4 -3 5 7 1 4 8 6 5 -7
9 15
10
4 -3 5 7 1 4 8 6 5 -7
-10 3
10
4 -3 5 7 1 4 8 6 5 -7
6 15
```

Sample output 1

```
2 1 1 2 3 5 4 3 5 4 3 6 7 7 6
-3 1 -7 4 5 4 5 7 8 6
4 -3 5 7 1 4 8 6 5 -7
-3 1 -7 4 5 7 4 8 6 5
4 -3 5 7 1 4 8 6 5 -7
-3 1 -7 4 5 7 4 8 6 5
4 -3 5 1 4 5 -7 7 8 6
```

Problem information

Author: PRO1

Generation: 2026-01-25T13:36:46.061Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>