

Ordena un Vector V

Y80100_es

Un **vector V** es un vector que está compuesto de dos partes:

$$v = x_1 \ x_2 \ x_3 \ x_4 \dots x_n \ y_1 \ y_2 \ y_3 \dots y_m$$

tales que $x_1 \dots x_n$ está ordenado de forma escrita **decreciente** y $y_1 \dots y_m$ está ordenado de forma escrita **creciente**. Además, $x_n > y_1$. Por último tenemos que $n, m > 0$. Es decir, ninguna de las dos partes está vacía.

Es necesario implementar la **función**

```
void ordena(const vector<int>& v, int pos, vector<int>& r)
```

con la siguiente especificación:

PRE : v es un vector V como $|v| \geq 3$, pos es la posición en la que se encuentra y_1 en v y $|v|=|r|$.

POST : El vector r contiene todos los elementos del vector v y está ordenado.

Observación

Sólo tiene que enviar la función que le pedimos y las acciones y funciones que vosotros mismos defina.

No se puede utilizar la operación `sort` de la biblioteca `stl`.

Pista: saber la posición en la que se encuentra y_1 le puede ayudar a ordenar el vector en tiempo lineal.

IMPORTANTE: Sólo hay que enviar la función que se pide, y quizás otras acciones y funciones necesarias. Hay que mantener, sin embargo, las definiciones de tipos y los `#includes`.

Entrada

Un número indeterminado de vectores V con el siguiente formato: un entero que indica su tamaño, después el vector V, después la posición en la que se encuentra y_1 en v y un vector r del mismo tamaño que v . Todo vector V tiene un tamaño mayor o igual a 3.

Salida

El vector r está ordenado y contiene todos los elementos de v .

Ejemplo de entrada 1

```
10  
20 18 16 2 4 6 8 10 12 14  
3
```

```
10  
20 2 4 6 8 10 12 14 16 18  
1
```

```
10  
20 18 16 2 4 6 8 10 12 14  
3
```

```
5  
14 11 8 2 14  
3  
  
10  
20 2 4 6 8 10 12 14 16 18  
1
```

Ejemplo de salida 1

```
2 4 6 8 10 12 14 16 18 20  
2 4 6 8 10 12 14 16 18 20
```

2 4 6 8 10 12 14 16 18 20
2 8 11 14 14
2 4 6 8 10 12 14 16 18 20

Información del problema

Autoría: PRO1

Generación: 2026-01-25T23:07:24.586Z

© Jutge.org, 2006–2026.

<https://jutge.org>