

## Vector Swap

จงเขียน function `vector_swap(v1, v2, s1, e1, s2, e2)` ซึ่งจะทำการสลับข้อมูลใน vector สองตัวคือ `v1` และ `v2` โดยสลับให้ข้อมูลตั้งแต่ช่องหมายเลข `s1` ถึงช่องหมายเลข `e1 - 1` ของ `v1` นั้นย้ายไปอยู่แทนที่ตำแหน่งของข้อมูลใน `v2` ตั้งแต่ช่องหมายเลข `s2` ถึงช่องหมายเลข `e2 - 1` และนำข้อมูลดังกล่าวใน `v2` ย้ายมาแทนที่ตำแหน่งของข้อมูลใน `v1` ที่ย้ายไป

รับประกันว่า  $0 \leq s1 \leq e1 \leq v1.size()$  และ  $0 \leq s2 \leq e2 \leq v2.size()$

ให้ดูตัวอย่างจากตารางด้านล่างนี้ เมื่อกำหนดให้ `v1 = <0, 1, 2, 3, 4>` และ `v2 = <50, 51, 52, 53, 54>`

s1	e1	s2	e2	ผลลัพธ์
0	2	0	2	<code>v1 = &lt;50, 51, 2, 3, 4&gt;</code> <code>v2 = &lt;0, 1, 52, 53, 54&gt;</code>
0	5	0	1	<code>v1 = &lt;50&gt;</code> <code>v2 = &lt;0, 1, 2, 3, 4, 51, 52, 53, 54&gt;</code>
2	4	4	5	<code>v1 = &lt;0, 1, 54, 4&gt;</code> <code>v2 = &lt;50, 51, 52, 53, 2, 3&gt;</code>
1	1	3	5	<code>V1 = &lt;0, 53, 54, 1, 2, 3, 4&gt;</code> <code>v2 = &lt;50, 51, 52&gt;</code>

กำหนดให้มี code เริ่มต้นให้เรียบร้อยแล้วแสดงด้านล่าง

### ข้อกำหนดเพิ่มเติม

ให้นิสิตเขียน code เฉพาะใน function `vector_swap` เท่านั้น โดยห้ามแก้ไข code ที่อยู่ใน function อื่น ๆ รวมถึง function `main` ด้วย ระบบ grader จะไม่ทำการตรวจสอบว่ามีการแก้ไข code ในส่วนอื่น ๆ หรือไม่ โดยจะตรวจจากผลลัพธ์เพียงอย่างเดียว อย่างไรก็ตาม จะมีคนมาอ่าน code อีกครั้งหนึ่งว่ามีการแก้ไข code ในส่วนที่ห้ามไว้หรือไม่ นิสิตสามารถแก้ไข `v1`, `v2` หรือกำหนดตัวแปรอื่น ๆ ได้ตามสะดวก แต่เมื่อจบฟังก์ชันแล้ว ทั้ง `v1` และ `v2` ต้องมีค่าเป็นไปตามที่กำหนด

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

using namespace std;

void vector_swap(vector<int> &v1, vector<int> &v2,
    int start1, int end1,
    int start2, int end2) {

}
```

```
int main() {
    //read input
    int n,c;
    vector<int>v1,v2;
    cin >> n; //number of v1
    for (int i = 0;i < n;i++) {
        cin >> c;
        v1.push_back(c);
    }
    cin >> n; //number of v2
    for (int i = 0;i < n;i++) {
        cin >> c;
        v2.push_back(c);
    }
    int s1,e1,s2,e2; //position
    cin >> s1 >> e1 >> s2 >> e2;

    //call the function
    vector_swap(v1,v2,s1,e1,s2,e2);

    //display content of the stack
    cout << "v1 has " << v1.size() << endl;
    for (auto &x : v1) { cout << x << " "; }
    cout << endl;

    //display content of the stack
    cout << "v2 has " << v2.size() << endl;
    for (auto &x : v2) { cout << x << " "; }
    cout << endl;
}
```