

## Vector Multiple Add

จงเขียนฟังก์ชัน `multiple_add(vector<int> &v,int value, int position, int count)` ซึ่งเป็นฟังก์ชันที่ทำการแทรกข้อมูล `value` เข้าไป ณ ตำแหน่ง `position` ของ `vector v` เป็นจำนวน `count` ตัว ตัวอย่างเช่น ให้ `v = {10,20,30}` การเรียก `multiple_add(v,-1,1,3)` จะทำให้ `v` กลายเป็น `{10,-1,-1,-1,20,30}` นั่นเอง

รับประกันว่า การเรียกฟังก์ชันนี้ `value` จะมีค่าอยู่ในช่วง `[-1000,1000]` ส่วน `position` จะมีค่าตั้งแต่ 0 ถึง `v.size()` (รวมทั้ง 0 และ `v.size()`) และ `count` นั้นมีค่าได้ตั้งแต่ 0 ถึง 1,000,000 ส่วน ขนาดของ `v` ที่ส่งให้ฟังก์ชันนั้นจะมีขนาดเป็นไป ได้ตั้งแต่ 0 ถึง 100

### ข้อบังคับ

ให้นิสิตเขียนฟังก์ชันนี้ลงในไฟล์ตามที่ระบุไว้ด้านล่างนี้ โดยห้ามนิสิตทำการแก้ไขไฟล์ในส่วนอื่น ๆ นอกเหนือไปจากในฟังก์ชันดังกล่าว และห้ามทำการพิมพ์ข้อมูลทางจอภาพหรืออ่านข้อมูลจากคีย์บอร์ดในส่วนที่เขียนเพิ่ม โดยเด็ดขาด (นั่นคือ ห้ามเพิ่มการเรียกใช้ `cin`, `cout`, `printf` หรือ `scanf` เป็นต้น นอกเหนือจากที่มีอยู่)

ระบบ grader จะไม่ทำการตรวจสอบว่านิสิตมีการแก้ไขฟังก์ชันหรือมีการอ่านเขียนข้อมูลหรือไม่ แต่จะมีคนมาอ่านคำตอบที่นิสิตส่งเข้ามา หากนิสิตทำผิดข้อกำหนดดังกล่าว จะถือว่าได้คะแนนเป็น 0 ทันที

### คำอธิบายฟังก์ชัน `main()`

โปรแกรมจะสร้าง `vector<int> v` ขึ้นมา และอ่านข้อมูลจากคีย์บอร์ดดังรูปแบบต่อไปนี้  
บรรทัดแรกประกอบด้วยจำนวนเต็มสี่ค่าคือ `n`, `value`, `position`, `count` โดย `n` ระบุถึงจำนวนข้อมูลใน `v` ตอนเริ่มต้น ส่วน `value`, `position`, `count` นั่นคือพารามิเตอร์ที่จะส่งให้ฟังก์ชัน `multiple_add`  
บรรทัดที่สองประกอบด้วยจำนวนเต็ม `n` จำนวนซึ่งระบุข้อมูลใน `v`  
`main()` จะอ่านข้อมูลดังกล่าว แล้วสร้าง `v` ขึ้นมาตามข้อมูลดังกล่าว และทำการเรียกฟังก์ชัน `multiple_add` หลังจากนั้นจะแสดงขนาดและข้อมูลของ `v` ออกทางหน้าจอ

### code เริ่มต้น

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

using namespace std;

void multiple_add(vector<int> &v, int value, int position, int count) {
    //write your code here
}

int main() {
    std::ios_base::sync_with_stdio(false);
    cin.tie(0);

    int n,value,position,count;
    cin >> n >> value >> position >> count;
    vector<int> v(n);
    for (int i = 0; i < n; i++) cin >> v[i];

    multiple_add(v,value,position,count);

    cout << "After call multiple_add" << endl << "Size = " << v.size() << endl;
    for (auto &x : v) cout << x << " ";
    cout << endl;
}
```

### ชุดข้อมูลทดสอบ

- 20% ของข้อมูลทดสอบจะมี count  $\leq 1$
- 30% ของข้อมูลทดสอบจะมี count  $\leq 100$
- 50% ของข้อมูลทดสอบจะมี count  $\leq 1000000$