

เจาะ Vector

ในโจทย์ข้อนี้ให้นิสิตเขียนฟังก์ชัน `punch(vector<string> &v, vector<string>::iterator it, int k)` ซึ่งจะทำการลบข้อมูลออกจาก vector `v` ณ ตำแหน่ง `it` รวมถึงข้อมูลที่อยู่ก่อนหน้า `it` เป็นจำนวน `k` ตัว และ หลัง `it` เป็นจำนวน `k` ตัวด้วยเช่นกัน หากมีข้อมูล ณ ตำแหน่งเหล่านั้นอยู่

ตัวอย่างเช่น ถ้าหาก `v` มีค่าเป็น {"A", "B", "C", "D", "E", "F", "G", "H"} และทำการเรียก `punch(v, v.begin()+3, 2)` จะทำให้ `v` กลายเป็น {"A", "G", "H"}

ในอีกตัวอย่างหนึ่ง ถ้าหาก `v` มีค่าเป็น {"A", "B", "C", "D", "E", "F", "G", "H"} และทำการเรียก `punch(v, v.begin(), 1)` จะทำให้ `v` กลายเป็น {"C", "D", "E", "F", "G", "H"} ให้สังเกตว่าไม่มีการลบข้อมูลตัวที่อยู่ก่อนหน้า `it` เนื่องจากไม่มีข้อมูลดังกล่าวอยู่

ในโจทย์ข้อนี้จะมีไฟล์ตั้งต้น `punch.cpp` ให้แล้ว ในไฟล์ดังกล่าวจะมีโครงของฟังก์ชัน `punch` อยู่ ให้เขียนโปรแกรมเพิ่มเติมในฟังก์ชัน `punch` และสามารถเพิ่มฟังก์ชันอื่นเข้าไปได้ แต่ห้ามแก้ไขฟังก์ชัน `main` โดยเด็ดขาด และห้ามเพิ่มหรือเปลี่ยนการแสดงผลข้อมูลออกทางหน้าจอจากไฟล์ที่ให้

ขอบเขตของข้อมูล

ในการเรียก `punch` แต่ละครั้ง ค่าของ `v`, `it` และ `k` จะเป็นไปตามเงื่อนไขต่อไปนี้

- ขนาดของ `v` จะไม่เกิน 100 และไม่น้อยกว่า 1
- $1 \leq k \leq 100$
- `it` จะระบุถึงสมาชิกใน `v` อย่างแน่นอน

คำอธิบายไฟล์ตั้งต้น

`main` ของไฟล์ตั้งต้นจะรับข้อมูลเข้ามาทางคีย์บอร์ดตามรูปแบบต่อไปนี้

- บรรทัดแรกรับข้อมูลสามตัวคือ จำนวนข้อมูลใน `v`, หมายเลขของตำแหน่งของ `it`, ค่า `k`
- หลังจากนั้นอีก `n` บรรทัดจะเป็นข้อมูลของ `v`

เมื่อโปรแกรมรับข้อมูลมาแล้ว จะเรียกฟังก์ชัน `punch` โดยใช้ข้อมูลที่รับมาจากคีย์บอร์ด และพิมพ์ผลลัพธ์ของการเรียกฟังก์ชัน `punch` ออกมาทางจอภาพ ตารางด้านล่างนี้แสดงตัวอย่างผลการทำงานของโปรแกรมที่ถูกต้อง

ตัวอย่าง

ข้อมูลนำเข้า	ข้อมูลส่งออก
5 1 1	DD
AA	EE
BB	
CC	
DD	
EE	

7 6 3	AA
AA	BB
BB	CC
CC	
DD	
EE	
FF	
GG	

ไฟล์ตั้งต้น punch.cpp

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

using namespace std;

vector<string> punch(vector<string> &v, vector<string>::iterator
it,int k) {
    //write some code here

    //don't forget to return something
}

int main() {
    int n,j,k;
    cin >> n >> j >> k;
    vector<string> v(n);
    for (int i = 0;i < n;i++) {
        cin >> v[i];
    }

    cout << "Result after punch" << endl;
    vector<string> result = punch(v, v.begin() + j, k);
    for (auto &x : result) {
        cout << x << endl;
    }
}
```