

## Stack Concat

จงเขียน function `stack_concat(s1, s2)` ซึ่งต้องทำให้ข้อมูลใน stack `s2` ทั้งหมดนั้นไปอยู่ "ด้านล่าง" ของ stack `s1` ตามลำดับที่ `s2` เป็น และทำให้ข้อมูลใน `s2` หายไปทั้งหมด ตัวอย่างเช่น สมมติให้ `s1` เป็น `<10, 20, 30, 40>` (โดยตัวบนสุดของ stack คือ 10, ตัวที่อยู่ถัดจากตัวบนสุดคือ 20,...) และ `s2` เป็น `<1, 2, 3>` หลังจากเรียก `stack_concat(s1, s2)` แล้ว `s1` จะต้องกลายเป็น `<10, 20, 30, 40, 1, 2, 3>` และ `s2` กลายเป็น `<>`

กำหนดให้มี code เริ่มต้นให้เรียบร้อยแล้วแสดงด้านล่าง

### ข้อกำหนดเพิ่มเติม

ให้นิสิตเขียน code เฉพาะใน function `stack_concat` 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 `s1, s2` หรือกำหนดตัวแปรอื่น ๆ ได้ตามสะดวก แต่เมื่อจบฟังก์ชันแล้ว ทั้ง `s1` และ `s2` ต้องมีค่าเป็นไปตามที่กำหนด

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

using namespace std;

void stack_concat(stack<int> &s1, stack<int> &s2) {
    //write your code only in this function
}

int main() {
    //read input
    int n,m;
    int c;
    cin >> n >> m;
    stack<int> s1,s2;
    for (int i = 0;i < n;i++) {
        cin >> c;
        s1.push(c);
    }
    for (int i = 0;i < m;i++) {
        cin >> c;
        s2.push(c);
    }

    //call the function
    stack_concat(s1,s2);

    //display content of the stack
    cout << "S1 has " << s1.size() << endl;
    while (!s1.empty()) {
        cout << s1.top() << " ";
        s1.pop();
    }
```

```
}  
cout << endl;  
  
//display content of the stack  
cout << "S2 has " << s2.size() << endl;  
while (!s2.empty()) {  
    cout << s2.top() << " ";  
    s2.pop();  
}  
cout << endl;  
}
```