重要公告

一、為了因應學校防疫全面升級，所有課程均需採取遠距教學方式，修習進階程式設計課程的同學，**到學期結束之前一律居家撰寫老師所出的程式作業**，禁止到電腦教室(一)上課寫程式，而且還必須於上課前利用手機**掃描以下 QRcode(限當日有效)，登入防疫期間課程點名系統完成到課登記**，以便老師可以掌握修課同學出席狀況。

二、由於同學可能在校外使用即時評測平台，然平台只接受校內 IP 上傳,所以**居住校外的同學**要將寫好的程式送上評測平台時要先設定 VPN 連線，詳細 VPN 連線設定詳如課程公告 **1100517 VPN 連線設定**內的 PPT 附檔所述，請同學自行參考運用，然因學校設備限制，最多只能 200 人同時 VPN 連線到本校，近日因疫情影響，學校行政人員居家上班的人數暴增，電子公文系統一定要使用 VPN 連線，所以 VPN 連線人數經常會超過，所以沒有要上傳程式到評測平台時，請 disconnect VPN 連線，要

用時再連線。。

三、日後不接受完成程式當面驗收方式，唯一的驗收方式是將評測平台通過的畫面截圖，截圖內容須包含完成日期時間、學號、題號、評測紀錄等資訊，上傳到雲端學院課程的該題討論版內，讓助教可以隨時去驗收，未完成留言驗收或延遲完成該動作，將依延遲時間長短酌減該程式分數 10~50 分不等。

四、老師的 WebEX 個人會議室（網址 ：https://moe-tw.webex.com/meet/hsiaojy)也會於課程期間開啟，請同學若有需要，可用完整姓名登入會議室內以便可以即時發問問題或討論。

五、若有題目需說明解題方法，會事先告知並在 WebEX 會議室內線上說明，相關說明文件及影音檔案也會在雲端學院公告，方便同學下載閱讀運用。

進階程式設計課程 程式設計作業#17

（請使用 C 或 C++語言撰寫解決下列問題之程式）

Advisor

Problem Description

It is hard to satisfy all our friends in life. But in more official scenarios, fairness must be achieved. Often, we use order of preferences and a deterministic process to compute a best-case scenario for all parties. In our problem, we will try to compute a match between some students and some advisors, both groups will have their order of preference. The process is detailed below,

1. Students' wishes come first.
2. If students' wishes conflict (e.g. two students want a professor equally much), they are settled with the professor's wishes.

We will use an example data of 4 by 4 (4 students and 4 professors),

```
S0: 3 2 1 0        # The student prefer professor 3 more than 2, and so on
S1: 3 2 0 1
S2: 0 1 2 3
S3: 0 2 3 1
P0: 0 1 2 3
P1: 0 1 2 3
P2: 0 1 2 3
P3: 0 1 2 3
```

Now we start the process, we'll work round by round, looking at the n^{th} column (n^{th} preferred professor) of each student at a time.

1. We look at the 1^{st} column of the students' wishes, we have (P3, P3, P0, P0), since two students want P0, and the other two want P3, we must resolve conflicts.

   A. Both S0 and S1 want P3, and P3 prefers S0 to S1, so we settle with (S0, P3)
   B. Both S2 and S3 want P0, and P0 prefers S2 to S3, so we settle with (S2, P0)

   After this round, S0 and S2 are out, P0 and P3 are occupied, so we no longer consider them in the rest of the matching process.
2. We look at the 2^{nd} column of S1 and S3, which is (P2, P2). Again, we'll look at the preference of P2, and we settle with: (S1, P2). After this round, S1 is out, P2 is occupied.
3. We look at the 3^{rd} column of S3, which is P3, who is occupied, continue.
4. We look at the 4^{th} column of S3, which is P1, we settle with: (S3, P1)

The result is: (S0, P3), (S1, P2), (S2, P0), (S3, P1).

Input

Each test case spans multiple finite lines.
The first line will consist of two positive integers $1 \leq N \leq M \leq 1000$ where N is the number of students, and M is the number of professors.
The next N lines will each represent a student, named from 0 to $N-1$. Each line consists of M distinct integers in $[0,M)$, which is the preference of the student, from most preferred to least.
The next M lines will each represent a professor, named from 0 to $N-1$. Each line consists of N distinct integers in $[0,N)$, which is the preference of the professor, from most preferred to least.

Output

For each test case, output a line consisting of the matching result, in the form of N integers, where the n^{th} (from 0) integer is the matched professor of the n^{th} student.

Sample Input

```
4 4
3 2 1 0
3 2 0 1
0 1 2 3
0 2 3 1
0 1 2 3
0 1 2 3
0 1 2 3
0 1 2 3
```

Sample Output

```
3 2 0 1
```