Simple Party Finding

Minions Legends is a multiplayer online battle arena (MOBA) mobile game. In this game, each player will be grouped with another player to form a team. The team will fight another team to defend their sacred ancient and to destroy their opponent's base.

In the next patch, the developers of the game want to create a simple party finding algorithm which can match two players based on their matchmaking rating (MMR). In their implementation, a player will be grouped with the player with the lowest MMR that is higher than their MMR. If a player has the highest MMR, they will be grouped with the player with the highest MMR that is lower than their MMR.

Format Input

The first line contains integer K which denote as the number of cases. The second line contains integer N stating the total number of players. Then, the third line consists of N MMR in the game. Lastly, the fourth line contains of F which denote as the player's MMR who wants to find a party. It is guaranteed that the MMR are distinct.

Format Output

Output with a format "CASE #K: [MMR_1] [MMR_2]"; where MMR_1 and MMR_2 are the result of party finding. Note that MMR_1 is the one with the lower MMR compared to MMR_2. If the inputted MMR's number is not in the list of players, give an output "CASE #K: -1 -1".

Constraints

- $1 \leq K \leq 100$
- $2 \leq N \leq 1000$
- $3000 \leq MMR \leq 9999$
- $3000 \leq F \leq 9999$

Sample Input 1 (standard input)

```
3
10
3246 4255 4873 5004 5095 6001 6692 7100 7610 9712
9712
8
4957 5444 5464 5634 6149 6622 8313 9492
4957
6
5393 6021 7233 8115 8118 9304
9999
```

Sample Output 1 (standard output)

```
CASE #1: 7610 9712
CASE #2: 4957 5444
CASE #3: -1 -1
```

Simple Party Finding

Minion Legends adalah sebuah permainan mobile multiplayer online battle arena (MOBA). Dalam permainan ini, setiap pemain akan dipasangkan dengan pemain lain untuk membuat sebuah tim; Tim tersebut akan melawan tim lain untuk menjaga sacred ancient mereka dan menghancurkan markas lawan.

Dalam patch selanjutnya, para pengembang permainan ini ingin membuat sebuah algoritma sederhana untuk mencari party yang dapat memasangkan dua pemain berdasarkan matchmaking rating (MMR) mereka. Dalam implementasi mereka, seorang pemain akan dipasangkan dengan pemain lain yang memiliki MMR terkecil yang lebih tinggi dari MMR mereka. Jika seorang pemain memiliki MMR terbesar, dia akan dipasangkan dengan pemain yang memiliki MMR terbesar yang lebih rendah dari MMRnya.

Format Input

Baris pertama berisi integer K yang menunjukkan jumlah testcase. Baris kedua berisi integer N yang menunjukkan jumlah pemain. Baris ketiga berisi N buah MMR dalam permainan. Terakhir, baris keempat berisi F, yaitu MMR pemain yang ingin mencari party. Dipastikan semua MMR unik.

Format Output

Output dengan format "CASE #K: [MMR_1] [MMR_2]"; dimana MMR_1 dan MMR_2 adalah hasil dari pencarian party, dimana MMR_1 adalah MMR pemain yang lebih rendah dari MMR_2. Jika MMR yang diinput tidak ada dalam list MMR pemain, berikan output "CASE #K: -1 -1".

Constraints

- $1 \leq K \leq 100$
- $2 \leq N \leq 1000$
- $3000 \leq MMR \leq 9999$
- $3000 \leq F \leq 9999$

Sample Input 1 (standard input)

```
3
10
3246 4255 4873 5004 5095 6001 6692 7100 7610 9712
9712
8
4957 5444 5464 5634 6149 6622 8313 9492
4957
6
5393 6021 7233 8115 8118 9304
9999
```

Sample Output 1 (standard output)

```
CASE #1: 7610 9712
CASE #2: 4957 5444
CASE #3: -1 -1
```