

# 숫자 전쟁

인하왕국과 비룡왕국은 전쟁 중이다. 두 나라의 각 병사들의 능력치는 양의 정수로 나타낼 수 있다. 능력치가  $x$ 인 병사는 능력치가  $x$ 의 배수인 병사만 무찌를 수 있다. 예를 들어, 능력치가 5인 병사는 능력치가 3, 9, 14인 병사는 무찌를 수 없지만, 능력치가 5, 10, 15, 12465인 병사는 무찌를 수 있다.

두 왕국의 왕은 용병을 1명씩 고용하려고 한다. 용병들도 병사들처럼 능력치를 양의 정수로 나타낼 수 있으며, 모든 양의 정수  $i$ 에 대해서 능력치가  $i$ 인 용병은 반드시 존재한다. 그런데 용병들은 상대 병사가 어느 왕국의 병사인지 구분할 수 없기 때문에, 자신이 무찌를 수 있는 병사는 소속 왕국을 가리지 않고 무조건 무찌른다. 따라서, 두 왕국의 왕은 자국의 병사들은 무찌르지 않되 상대 왕국의 모든 병사를 무찌를 수 있는 용병을 고용하려고 한다. 즉, 인하왕국에서 고용한 용병은 인하왕국의 병사는 아무도 무찌를 수 없으며, 비룡왕국의 모든 병사를 무찌를 수 있어야 한다. 또한, 비룡왕국에서 고용한 용병은 비룡왕국의 병사는 아무도 무찌를 수 없으며, 인하왕국의 모든 병사를 무찌를 수 있어야 한다. 이러한 조건을 만족하는 용병이 여러 명 존재한다면, 능력치가 가장 높은 용병만을 고용한다.

예를 들어, 인하왕국의 병사들의 능력치가 각각 4, 10, 14이며, 비룡왕국의 병사들의 능력치가 각각 12, 6, 54인 경우에 대해 알아보자. 이때, 인하왕국에서 능력치가 3인 용병을 고용하면, 해당 용병은 비룡왕국의 모든 병사를 무찌를 수 있으며, 인하왕국의 병사는 아무도 무찌를 수 없다. 또한, 능력치가 6인 용병도 비룡왕국의 모든 병사를 무찌를 수 있으며, 인하왕국의 병사는 아무도 무찌를 수 없다. 이러한 조건을 만족하는, 능력치가 6보다 큰 용병은 존재하지 않는다. 따라서 인하왕국은 능력치가 6인 용병을 고용한다. 하지만, 비룡왕국은 이와 같은 조건을 만족하는 용병이 존재하지 않으므로, 용병을 고용할 수 없다.

인하왕국과 비룡왕국의 병사들의 능력치가 주어질 때, 두 왕국이 고용할 용병의 능력치를 공백으로 구분하여 출력한다. 단, 고용할 수 있는 용병이 없다면 -1을 출력한다.

※ 프로그램의 실행 시간은 1 초, 메모리 사용량은 512MB 를 초과할 수 없다.

사용할 수 있는 언어는 C, C++로 제한한다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);
```

단, 위의 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력을 혼용해서 사용하면 안된다. C++의 std::endl 의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n";을 사용하는 것을 권장한다.

# 입력

첫 번째 줄에는 테스트 케이스 수  $T$  ( $1 \leq T \leq 100$ )가 주어진다.

이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에 인하 왕국과 비룡 왕국의 병사들의 수  $n, m$  ( $1 \leq n, m \leq 100,000$ )이 공백으로 구분되어 주어진다.
- 두 번째 줄에 인하 왕국  $n$ 명의 병사들의 능력치가 공백으로 구분되어 주어진다.
- 세 번째 줄에 비룡 왕국  $m$ 명의 병사들의 능력치가 공백으로 구분되어 주어진다.

병사들의 능력치는 100,000,000보다 작거나 같은 양의 정수이다.

# 출력

각 테스트 케이스마다 인하왕국과 비룡왕국이 고용할 용병의 능력치를 공백으로 구분하여 한 줄에 출력한다. 고용할 수 있는 용병이 없다면 -1을 출력한다.

## 예제 입출력

예제 입력	예제 출력
2	6 -1
3 3	107 5
4 10 14	
12 6 54	
5 1	
50 10 25 30 400	
107	