

Week 12 - Hashing

- 주의 사항: 부정행위 금지(채점서버 외 인터넷 사용금지), STL 사용금지 (string, vector는 사용 가능)
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

문제 2

해시 테이블을 탐색하고 탐색의 성공 여부와 탐색횟수(probing)을 출력해보자. 배열의 크기가 P ($2 \leq P \leq 1,000$)인 해시 테이블을 작성하고, 해시 함수를 통해 해시 테이블에 엔트리(E_{key}, E_{value})를 삽입한다. 그리고 탐색할 Q_{find} ($1 \leq Q_{find} < P$)개의 자연수 R ($1 \leq R \leq 1,000$)이 주어진다. 이 S 개의 자연수를 주어진 해시 테이블에서 탐색한다. 이 때 해시 테이블은 다음 조건을 만족한다.

1. 충돌이 일어날 경우, **선형 조사법(Linear Probling)**을 사용하여 충돌을 처리한다.
2. 해시 함수는 그림 1의 해시 함수를 사용하기로 한다. 그림 1의 해시 함수는 입력 받은 자연수 key ($1 \leq key \leq 1,000$)를 배열의 크기 P 로 나머지 연산하는 함수이다.
3. 해시 테이블 생성 시 모든 배열의 엔트리의 키(E_{key})와 값(E_{value})은 -1로 초기화한다.

```
int hashfunc(int key)
{
    return key % P;
}
```

그림 1. 해시 함수

입력

- 1. 첫 번째 줄에는 테스트 케이스의 수 T 가 주어진다. ($1 \leq T \leq 1,000$)
- 2. 두 번째 줄부터, 다음이 T 번 반복된다.
 - 1. 해시 테이블의 배열의 크기 P ($2 \leq P \leq 1,000$)가 주어진다.
 - 2. 해시 테이블에 삽입할, 엔트리의 키(E_{key})와 값(E_{value})이 될 자연수 key ($1 \leq key \leq 1,000$)의 개수 Q_{put} ($1 \leq Q_{put} < P$)가 주어진다. (따라서, $key = E_{key} = E_{value}$)
 - 3. Q_{put} 개의 자연수 key 가 빈칸을 사이에 두고 주어진다. (단, 값은 중복되지 않는다)
 - 4. 해시 테이블에서 탐색할, 자연수 R ($1 \leq R \leq 1,000$)의 개수 Q_{find} ($1 \leq Q_{find} < P$)가 주어진다
 - 5. Q_{find} 개의 자연수 R 이 빈칸을 사이에 두고 주어진다. (단, 값은 중복되지 않는다)

출력

매 테스트 케이스마다 주어진 해시 테이블에서, 자연수 R 이 입력된 순서대로 다음을 처리한다.

탐색 연산을 하여 R 과 일치하는 해시 테이블의 엔트리의 키(E_{key})를 탐색하는데 성공하면 "True"와 탐색횟수(probing)을 빈 칸을 사이에 두고 출력한다. 존재하지 않으면 "False"와 탐색횟수(probing)을 빈칸을 사이에 두고 출력한다.

예제 입출력

예제 입력	예제 출력
2	True 2
7	False 4
3	True 1
15 8 2	True 2
4	True 1
8 22 15 2	True 2
29	False 2
9	False 3
78 9 67 45 90 32 83 18 12	False 1
5	
9 67 70 3 17	