

〈알고리즘 실습〉 - 해시테이블

※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 특별한 언급이 없으면, 각 줄의 맨 앞과 맨 뒤에는 공백을 출력하지 않는다.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 입출력 예시에서 \mapsto 이 후는 각 입력과 출력에 대한 설명이다.

[문제 1] (분리연쇄법 해시테이블) 크기 M 인 해시테이블에 여러 개의 키 값을 입력받아 저장하고, 분리연쇄법을 이용하여 충돌을 처리하는 해시테이블 프로그램을 작성하시오.

○ 구현 조건

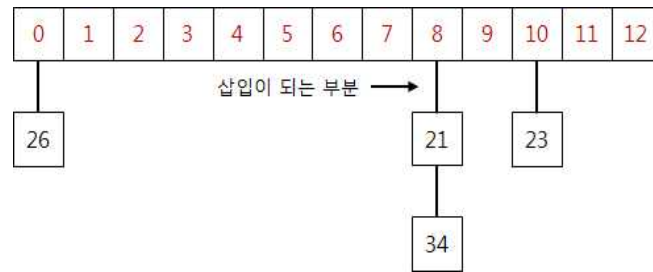
- 해시테이블은 크기가 M 인 배열로 동적 할당한다.
- 입력 키는 중복이 없는 자연수다.
- 키 x 에 대한 해시함수 $h(x) = x \% M$ 을 사용한다.
- 삽입 시 충돌이 발생하는 경우, 해당 버킷 리스트의 맨 앞에 삽입한다.

○ 입력

- 해시테이블의 크기 M 을 입력받는다.
- 삽입(**i**), 탐색(**s**), 삭제(**d**), 인쇄(**p**) 명령어를 순서에 상관없이 반복하여 입력받는다.
 - i** < x > : 키 x 를 해시테이블에 삽입
 - s** < x > : 키 x 가 해시테이블에 존재하는지 탐색
 - d** < x > : 키 x 가 해시테이블에 존재하면 삭제
 - p** : 해시테이블에 저장된 키들을 순서대로 인쇄(입출력 예시 참조)
 - e** : 프로그램 종료

○ 출력

- 키를 삽입하였을 때 아무 출력을 하지 않는다.
- 탐색 또는 삭제할 때, 키가 존재하면 리스트에서 해당 키가 저장된 순위(1부터 시작)를 인쇄하고 없다면 0을 인쇄한다(해시테이블의 주소가 아닌 리스트에서의 순위를 인쇄함에 유의).
- 해시테이블을 인쇄할 때, 0번 주소부터 마지막 주소까지 순회하면서 저장된 키들을 방문하는 순으로 인쇄한다.



입력 예시 1

출력 예시 1

13	↳ M		
i 34	↳ 34 삽입		
i 23	↳ 23 삽입		
i 26	↳ 26 삽입		
i 21	↳ 21 삽입		
s 36	↳ 36 탐색		
s 26			
s 34			
s 21			
p	↳ 해시테이블 인쇄		
d 21	↳ 21 삭제		
s 34			
d 8			
e			

0	↳ s 36 탐색 결과
1	↳ s 26 탐색 결과
2	↳ s 34 탐색 결과
1	↳ s 21 탐색 결과
□26 21 34 23	↳ p 입력 결과
1	↳ d 21 삭제 결과
1	↳ s 34 탐색 결과
0	↳ d 8 삭제 결과

[문제 2] (개방주소법 해시테이블 – 선형조사법) 크기 **M**인 해시테이블에 **n**개의 키 값을 입력받아 저장하고, 개방주소법 중 **선형조사법**을 이용하여 충돌을 처리하는 해시테이블 프로그램을 작성하시오.

- 구현 조건
 - 해시테이블은 크기가 **M**인 배열로 동적 할당한다.
 - **n**은 **M**보다 작은 자연수로 최대 삽입 개수다.
 - 입력 키는 중복이 없는 **6**자리 또는 **8**자리의 임의의 자연수(학번)다.
 - 키 **x**에 대한 해시함수 $h(x) = x \% M$ 을 사용한다.
 - 저장된 키 값이 없는 빈 버킷은 **0**으로 처리한다.
- 입력
 - 해시테이블의 크기 **M**과 입력 데이터의 크기 **n**을 입력받는다.
 - 삽입(**i**), 탐색(**s**) 명령어를 순서에 상관없이 반복하여 입력받는다.
 - i <x>** : 키 **x**를 해시테이블에 삽입
 - s <x>** : 키 **x**가 해시테이블에 존재하는지 탐색
 - e** : 프로그램 종료
- 출력
 - 키를 삽입하였을 때, 저장된 주소(배열 인덱스)를 인쇄한다.

- 삽입할 때 충돌이 일어나면 선형조사법에 의해 다음 셀로 이동하여 충돌 검사를 계속한다. 충돌 횟수만큼 **C**를 인쇄한 후, 삽입에 성공한 주소를 인쇄한다.
- 탐색한 키가 테이블에 존재할 경우, 키가 저장된 주소와 값을 인쇄한다(없을 경우 **-1**을 인쇄한다).

입력 예시 1

```
7 3          ↳ M n
i 17011112   ↳ 삽입
i 17012345
i 17012687
s 17011111   ↳ 탐색
e
```

출력 예시 1

```
6          ↳ 17011112의 주소
0          ↳ 17012345의 주소
CC1        ↳ 17012687의 주소(충돌 2회)
-1         ↳ 17011111 탐색 결과(없음)
```

입력 예시 2

```
13 10
i 16110243
i 17011111
i 17012331
i 17012354
i 17013672
i 16012342
s 17012354
i 15013986
i 102067
i 113478
i 14012322
s 16110243
e
```

출력 예시 2

```
6          ↳ 16110243 주소
0          ↳ 17011111 주소
11         ↳ 17012331 주소
8          ↳ 17012354 주소
C1         ↳ 17013672 주소(충돌 1회)
C9         ↳ 16012342 주소(충돌 1회)
8 17012354 ↳ 17012354 탐색 결과(주소, 키)
CC2        ↳ 15013986 주소(충돌 2회)
4          ↳ 102067 주소
CC3        ↳ 113478 주소(충돌 2회)
12         ↳ 14012322 주소
6 16110243 ↳ 16110243 탐색 결과(주소, 키)
```

[문제 3] (개방주소법 해시테이블 – 이중해싱) 문제 2에서 충돌처리 방법을 이중해싱으로 변경하시오.

- 구현 조건
 - 해시테이블은 크기가 **M**인 배열로 동적 할당한다(종료 시 해제).
 - **n**은 **M**보다 작은 자연수로 최대 삽입 개수다.
 - 입력 키는 중복이 없는 2자리 이상의 자연수다.
 - 키 **x**에 대한 첫 번째 해시함수 $h(x) = x \% M$, 두 번째 해시함수 $h'(x) = q - (x \% q)$ 를 사용한다. **q**는 **M**보다 조금 작은 소수로 입력으로 주어진다.
 - 저장된 키가 없는 빈 버킷은 **0**으로 처리한다.
- 입력
 - **M, n, q**를 입력받는다.
 - 삽입(**i**), 탐색(**s**), 출력(**p**) 명령어를 순서에 상관없이 반복하여 입력받는다.
 - i <x>** : 키 **x**를 입력받아 해시테이블에 삽입

s <**x**> : 키 **x**가 해시테이블에 존재하는지 탐색

p : 현재의 해시테이블 인쇄

e : 해시테이블 인쇄 후 프로그램 종료

○ 출력

- 키를 삽입하였을 때, 저장된 주소(배열 인덱스)를 인쇄한다.
- 삽입할 때 충돌이 일어나면 두 번째 해시함수로부터 얻은 셀로 이동하여 충돌 검사를 계속한다. 충돌 횟수만큼 **C**를 인쇄한 후, 삽입에 성공한 주소를 인쇄한다.
- 탐색한 키가 테이블에 존재할 경우, 키가 저장된 주소와 값을 인쇄한다(없을 경우 **-1**을 인쇄한다).

입력 예시 1

출력 예시 1

13 10 11	↳ M n q		
i 25	↳ 25 삽입	12	↳ 25 주소
i 13		0	↳ 13 주소
i 16		3	↳ 16 주소
i 15		2	↳ 15 주소
i 70		5	↳ 70 주소
p	↳ 출력	□13 0 15 16 0 70 0 0 0 0 0 0 25	↳ 해시테이블 출력
i 28		C7	↳ 28 주소(충돌 1회)
i 31		CC9	↳ 31 주소(충돌 2회)
i 20		CC11	↳ 20 주소(충돌 2회)
i 14		1	↳ 14 주소
s 20	↳ 20 탐색	11 20	↳ 20 탐색(주소, 키)
s 27		-1	↳ 27 탐색(없음)
i 38		CCC4	↳ 38 주소(충돌 3회)
e	↳ 종료	□13 14 15 16 38 70 0 28 0 31 0 20 25	↳ 해시테이블 인쇄