

<알고리즘 과제> 해시테이블

문제: 해시테이블 비활성화 방식 삭제의 C 프로그램 구현 (심층문제 12-2 참조)

비활성화 방식의 삭제를 구사하는 개방주소법의 관련 알고리즘을 C 프로그램으로 구현하시오.

- findElement(k)
- insertItem(k, e)
- removeElement(k)
- showHashTable(H):

아래 예시처럼 현재 해시테이블의 내용을 두 개의 라인으로 보여줄 것.

윗 라인은 해시테이블의 배열첨자, 아래 라인은 저장된 원소들임.

비어있는 셀은 점으로 표시할 것.

```
0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
.  . 16  .  . 8 21  .  . 42  .  . 30 15 29 55  .  .  . 9  .  . 37
```

- 상기의 4가지 이외에 추가적으로 필요한 메소드는 정의하여 이용할 것.
- 함수 인수(Parameter)의 갯수와 형태(Type)는 원하는 대로 바꾸어도 됨.

프로그램 요구사항:

- 탐색, 삽입, 삭제를 하나의 프로그램으로 묶어서 구현할 것
- main 함수에서 "반복적으로" 사용자의 명령코드 f(탐색), i(삽입), r(삭제)에 따라 해당 메소드를 호출하여 처리하는 방식으로 작성할 것
- 예를 들어, "f 14" 명령은 findElement(14)를 호출, "i 25 4" 명령은 insertItem(25,4)를 호출, "r 25" 명령은 removeElement(25)를 호출
- q(quit) 명령이 입력되면 프로그램 종료
- 해시테이블의 사이즈 M = 23 으로 정의할 것.
- 이중해싱에 기반해야 하며 h 및 h'은 각자 적절히 정의하여 사용할 것.
- 저장 원소는 0에서 99 사이의 정수로 제한할 것.
- 삽입(i) 명령시 중복 키 또는 해시테이블이 만원일 경우 적절한 안내 메시지와 함께 명령 수행을 거절할 것.
- 탐색(f) 또는 삭제(r) 명령시, 함께 주어진 키가 존재하지 않을 경우 NoSuchKey를 반환 및 인쇄할 것.
- 명령 코드 인수(Argument)의 갯수와 형태(Type)는 원하는 대로 바꾸어도 됨.

실행에 작성 지침:

- f 명령에 대해서는 탐색결과 원소를 출력해야 하며, i 혹은 r 명령을 수행한 직후에는 현재 해시테이블의 내용을 출력하여 방금 명령을 제대로 수행하였는지 눈으로 확인 가능하도록 할 것
- 탐색, 삽입, 삭제 명령을 각각 3회를 적당히 섞어서 실시하고 종료할 것
- 본인이 테스트한 명령의 순서를 코멘트에 넣어 채점자가 테스트가 가능하도록 할 것.
- 원만한 실행을 위해서 20개의 항목으로 해시테이블을 초기화한 후 명령코드를 입력받기 시작하

도록 작성할 것 ([힌트 참조](#))

힌트: 주요 함수들의 설계 가이드라인

global integer **MAX** = 23 {size of the hash table **H**}

Alg *main*()

1. **H** ← createHashTable() {Create hash table of size **MAX** with 20 initial element based on double hashing}
2. while(True)
 - (cmd, k, e) ← read() {read in command, key, element triple}
 - switch(cmd)
 - 'f': e ← findElement(k)
 write(e)
 - 'i': insertItem(k, e)
 showHashTable(H)
 - 'r': removeElement(k)
 showHashTable(H)
 - 'q' : exit
3. return