⟨14장 해싱 연습제; p569~570, 2019/0783 김성현⟩   h(b) =>해쉬 함수

p.569   01. h(b1)=h(b2) → 충돌(collision) ②

02. 나머지 $13\overline{)20}^{2}$ 27 → ①
①   130 → ⓪   (1, 0)   ⓒ
$\dfrac{10}{13\overline{)130}}$
$\dfrac{130}{0}$

03. h(k) = k mod 11 = k % 11   (1)

mod → %1

(1) table:
```
0 | 44
1 | 12
2 | 13
3 | 88
4 | 23
5 | 11
6 | 94
7 | 39
8 | 16
9 | 20
10| 5
```
선형조사

• 오버플로우(overflow)란?

→ 충돌이 버킷에 할당된 슬롯 수보다 많이 발생하는 것.

(4)
```
0 | 44 → 88 → 11
1 | 12 → 23
2 | 13
3 |
4 |
5 | 16 → 5
6 | 94 → 39
7 |
8 |
9 | 20
10|
```

재배조   (2)
놓여봐
```
0 | 44
1 | 12
2 | 13
3 | 16
4 | 88
5 | 23
6 | 94
7 | 39
8 | 5
9 | 11
10| 20
```
빈 조사

h'(k) = 7 - (k mod 7)

버킷=1 → 충돌 발생하면 발생시 과실없수 시름?

(3)
```
0 | 44
1 | 12
2 | 13
3 | 88
4 | 16
5 | 23
6 | 94
7 | 11
8 | 39
9 | 20
10| 5
```

$7\overline{)28}$
$-\dfrac{44}{80}$ $7\overline{)23}$
6   21
6   2
4

04.

☆ int hash_function (char *key) → 문자열 해시 테이블에 저장하기위해

                                      아스키코드값을 더한거 함수.
```
{
    int hash_index = 0;

    while (*key)
        hash_index = 2 * hash_index + *key++;
    return hash_index;
}
void hash_chain_add (element item, struct list *ht[]) // 체인열 사용 테이블 키(값)
{

    int hash_value = hash_function(item.key);
    struct list *ptr;
    struct list *node_before = NULL, *node = ht[hash_value];
    for (; node ; node_before = node , node = node → link ) {
        if(node → item.key == item.key) {
            fprintf (stderr, "이미 탐색키가 저장되어 있음 \n");
            return;
        }
    }

    ptr = (struct list *) malloc (sizeof (struct list);
    ptr → item = item;
    ptr → link = NULL;
```

p.570 05. (3)

06. 중간 제출함42

07. M: 해시테이블 크기 , k = 해시값
(1) $(3+i^2) \% 17$ for $i \geq 0$
   $i=0, 1, 2, 3, \rightarrow 0$ / $i=4 \rightarrow 2$ / $i=5 \rightarrow 11$

(2) $i=6$
   i) $(3+36) \% 17 \Rightarrow 39 \% 17 = 5$
   ii) $(k_6 + 13) \bmod 17$

(3)