【問5】 B=8，ハッシュ関数 h(x) = x % B（x を B で割った余り）としたとき，7 個の整数データ 11, 5, 13, 9, 3, 12, 6 を左から順に内部ハッシュ法ですべて挿入した後のハッシュ表を求めよ．ただし，挿入には以下に示す図 5 のプログラムを用いるものとし，挿入前のハッシュ表 A[] は空（すべての A[i].state == empty）であるとする．また，解答は図 6 に示すように配列 A の element の部分のみを示せばよい．(14 点)

```
enum oed {occupied, empty, deleted};  /* 列挙型データ oed */
struct hash {
  int element;
  enum oed state;  /* メンバ state は occupied, empty, */
};                 /* deleted のいずれかの値を取る      */

void insert(int x, struct hash A[])
{
  int i, j;

  i = h(x); j = 0;
  while(A[i].state == occupied && j < B){
    i = (i+1) % B;
    j++;
  }
  if(j == B){
    printf("The hash table is full!\n");
    return;
  }
  A[i].element = x;
  A[i].state = occupied;
}
```

図 5: 構造体 struct hash の定義と内部ハッシュ法による挿入関数 insert
（アルゴリズム自体は授業で説明した内部ハッシュ法そのものである．）

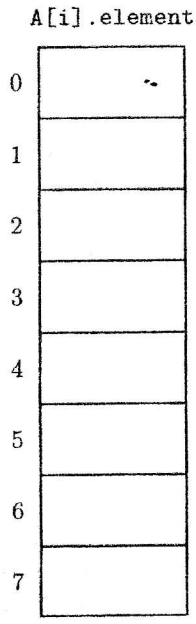

図 6: ハッシュ表

【問6】 8 個の整数データ 6, 4, 8, 3, 1, 5, 7, 2 を，次ページに示す図 7〜図 9 クイックソートのアルゴリズム（プリントの図 4.E〜図 4.G と同じアルゴリズム）を用いて整列することを考える．整列前の配列 A に上記のデータが順番に格納されている（つまり A[0]=6, A[1]=4, ..., A[7]=2 である）とき，関数 quicksort のすべての呼び出し時の配列 A の状態を呼び出しの順番どおりに以下のように記せ．(16 点)

```
quicksort([6,4,8,3,1,5,7,2],0,7)  <--- 最初の呼び出し
                ⋮
                ⋮
quicksort([1,2,3,4,5,6,7,8],7,7)  <--- 最後の呼び出し
```