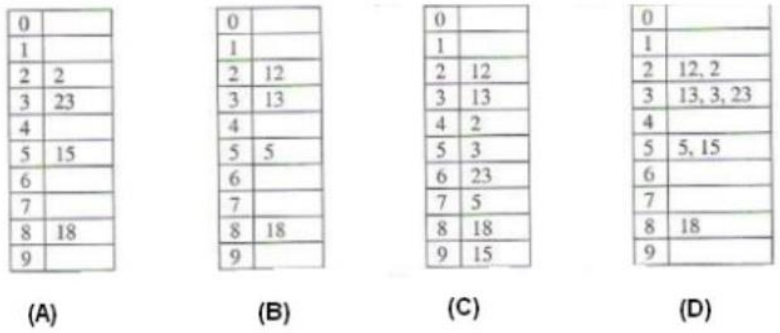
Part B: Hashing

1. Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true?

   i. 9679, 1989, 4199 hash to the same value
   ii. 1471, 6171 hash to the same value
   iii. All elements hash to the same value
   iv. Each element hashes to a different value

   (A) i only
   (B) ii only
   (C) i and ii only
   (D) iii or iv

2. The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table?



0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

(A)

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

(B)

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

(C)

0	
1	
2	12, 2
3	13, 3, 23
4	
5	5, 15
6	
7	
8	18
9	

(D)

3. A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

(A) 46, 42, 34, 52, 23, 33
(B) 34, 42, 23, 52, 33, 46
(C) 46, 34, 42, 23, 52, 33
(D) 42, 46, 33, 23, 34, 52

4. What is the problem if hash functions are implemented as type h(key)=key mod 10 or h(key)=key mod 20?
5. Assume the following data structure as a hash table:

```
struct DataItem {
   int data;
   int key;
};
```

And the hash function h(key)=key mod 20 with linear probing. Implement basic functions of the hash table:

- Search for a key in the hash table and return a pointer to the corresponding DataItem

```
struct DataItem *search(int key);
```

- Insert a new key-value pair (a new DataItem element) to the hash table

```
void insert(int key, int data);
```

- Remove a given DataItem element from the hash table

```
void remove(struct DataItem* item);
```

- Display the whole hash table

```
void display();
```

Assume that the hash table is defined as a static array of 20 pointers to the structure DataItem:

```
struct DataItem* hashArray[20];
```

Note: You can analyze, for example, the code from this web page:

https://www.tutorialspoint.com/data_structures_algorithms/hash_table_program_in_c.htm#