CS 2020 HW 14 **Hash Tables** **Name:**

```
#define MAX_CAPACITY    11
#define R      5
struct HashTable {
  int hashtable[MAX_CAPACITY];    // table to store values
  int count;                  // count of values currently stored
};
int hash1(int value) {
  return (value % MAX_CAPACITY);
}
int hash2(int value) {
  return (R - (value % R));
}
```

1) Use hash1 function and *linear probing* as a collision resolution strategy, insert the following elements into the correct index of the hash table: 1, 5, 4, 11, 21, 15, 22, 23.

```
set i to 0
while collision is detected
  index = ( hash1(elem) + i++ ) % MAX_CAPACITY;
hashtable[index] = elem
```

Elem											
Index	0	1	2	3	4	5	6	7	8	9	10

2) How many collisions were detected when inserting: a) 15 b) 21 c) 22 d) 23

3) Use hash1 and hash2 function (*double hashing*) and *linear probing* as a collision resolution strategy, insert the following elements into the correct index of the hash table: 1, 5, 4, 11, 21, 15, 22, 23.

```
set i to 0
while collision is detected
  index = ( hash1(elem) + (i++ * hash2(elem) ) % MAX_CAPACITY;
hashtable[index] = elem
```

Elem											
Index	0	1	2	3	4	5	6	7	8	9	10

4) How many collisions were detected when inserting: a) 15 b) 21 c) 22 d) 23