(4 points) Given the sequence $< 3, 10, 2, 4 >$, apply the double-hashing strategy for open addressing to store the sequence in the given order in a hash table of size m = 5 with hash functions h1(k) = k mod 5 and h2(k) = 7k mod 8. Document all collisions and how they are resolved. Write down your computations.

First insertion: 3

$h(3) = 3$

M	Value
0	
1	
2	
3	3
4	

Second insertion: 10

$h(10) = 10 \mod 5 = 0$

M	Value
0	10
1	
2	
3	3
4	

third insertion: 2:

$h(2) = 2 \mod 5 = 2$

M	Value
0	10
1	
2	2
3	3
4	

fourth insertion: 4

$h(4) = 4 \mod 5 = 4$

M	Value
0	10
1	

2	2
3	3
4	4

since no collisions occurred, the second hash function was not used
B) see hashtable.h for implementation, main.cpp for test program.
The hashing algorithm implemented is key % maxsize, this ensures that each address of the generated hash table can be directly accessed, and also ensures that there is enough space to store all of the inserted elements as long as the number of elements inserted is not greater than the max size. Upon collisions, the algorithm simply auto increments by 1 and uses the modulo operator to as to not overflow and also checks if there is no more space to insert.

Problem 10.2
(2 points) Show that a greedy algorithm for the activity-selection problem that makes the greedy choice of selecting the activity with shortest duration may fail at producing a globally optimal solution.
Consider 3 activities over a time span of 10 time units with time values:
A = [1-5] B = [4-7] C =[6-10], where the numbers correspond to the start and stop times of the activities.
Activity B has the shortest duration, with a duration of 3. However, if this event is selected, none of the remaining activities can be chosen. However, this is not the globally optimum solution, as this solution is {A, C} , which does not have any time clashes.