Hash Tables: Designing Hash Function

Division Method (or Remainder Method) Steps

1. We pick a collection of items.
2. Our hash function *h*, will take one item at a time and divide it by the number of slots.
3. The remainder from the division will be the hash value and will determine the slot number.

Hash Function *h(item) = item % m* (m = number of slots)

Item	Hash Value (slot number)
26	
35	
27	
17	
42	
70	
57	
45	
86	
4	

Once Hash values are calculated, let's put them in our hash table.

Index	Item	Collision Resolution with Linear Probing
0		
1		
2		
3		
4		
5		

6		
7		
8		
9		
10		

This hash table has several slots where multiple items are assigned.
Let's use **linear probing** to find new slots for these items. Use the other column of the table above to do it.

Exercises

Problem 1

In a hash table of size 13 which index positions would the following key pairs map to?

A. 1, 10
Answer

B. 13, 0
Answer

C. 1, 0
Answer

D. 2, 3
Answer

Problem 2

Suppose you are given the following set of keys to insert into a hash table that holds exactly 11 values:

113 , 117 , 97 , 100 , 114 , 108 , 116 , 105 , 99.

Which of the following best demonstrates the contents of the hash table after all the keys have been inserted using **open addressing with linear probing**?

A. 100, __, __, 113, 114, 105, 116, 117, 97, 108, 99

B. 99, 100, __, 113, 114, __, 116, 117, 105, 97, 108

C. 100, 113, 117, 97, 14, 108, 116, 105, 99, __, __

D. 117, 114, 108, 116, 105, 99, __, __, 97, 100, 113

Problem 3

Insert the keys 17, 3, 9, 39, 5, 6, 28, and 22 into a hash table of size 11 given using the hash function $h(x) = x \bmod 11$. Use the Chaining method.