1. Suppose that we use direct-access table $T$ to store a set of integer values. As these integers take value ranging from 0 and $m$, table $T$’s size is $m + 1$, and $T[i] = 1$ if value $i$ is in the set, and $T[i] = 0$ if value $i$ is not in the set.

   (a) Describe an algorithm to find the largest value in the set with a worst case running time of $T(m) = \Theta(m)$.

   (b) Explore ways to do better (i.e., come up a more efficient algorithm for the problem).
2. Demonstrate the insertion of the keys 3, 27, 18, 15, 20, 33, 12, 18, 9 into a hash table with collision resolved by chaining. Please draw the hashtable after all keys have been inserted. Let the table have 9 slots, and let the hash function be \( h(k) = k \mod 9 \).

3. Consider a hash table where the key’s type is char string. One way to interpret (or convert) string to integer value is to treat the string as a radix-128 number, as illustrated below:

String "river" is made up of chars 'r', 'i', 'v', 'e', 'r', each of which corresponding to an integer value between 0 and 127 according to ASCII code. Therefore we can treat the string as an integer represented in base 128, with its one’s digit value given by 'r' (114), its 128 digit’s value is 'e' (101), and so on.

So the whole string corresponds to the number: 114 \times 128^4 + 105 \times 128^3 + 118 \times 128^2 + 101 \times 128 + 114.

Answer the following questions:

(a) What’s the number corresponding to string ”bye”?

(b) Could there be two different strings being coverted into the same integer? Given at least one example.
(c) (Extra Credits) Suppose after the above conversion, we then use the division method to calculate the hash value. Suppose that the key value could be an arbitrarily long string (so its corresponding radix-128 number can be arbitrarily large), could you use a constant number of bytes (bits) to perform the conversion and calculate the hash value? Assume that the size of hash table $m$ is 64 bits long.
4. Consider a hash table of size \( m = 1000 \) and the hash function \( h(k) = \lfloor m(kA \mod 1) \rfloor \) for \( A = (\sqrt{5} - 1)/2 \). Compute the locations to which the keys 62, 63, 64, 65 are mapped. Note that:

- \( kA \mod 1 \) means to take the decimal part of \( kA \). For example, if \( kA = 246.1234 \), then \( kA \mod 1 = 0.1234 \). An equivalent way to express this is \( kA - \lfloor kA \rfloor \).
- \( \lfloor x \rfloor \) is defined to be the largest integer that is less than or equal to \( x \), with examples below: \( \lfloor 2.34 \rfloor = 2 \), \( \lfloor 12 \rfloor = 12 \), \( \lfloor -102.34 \rfloor = -103 \).

5. Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into a hash table of length \( m = 11 \) using open addressing with the primary hash function \( h'(k) = k \mod m \). Illustrate the result of inserting these keys using linear probing, using quadratic probing with \( c_1 = 1 \) and \( c_2 = 3 \), and using double hashing with \( h_2(k) = 1 + (k \mod (m - 1)) \).