1. **Exercise 11.3-4 (p. 236) (20 points)**

   Consider a hash table of size $m = 1000$ and a corresponding hash function $h(k) = [m(kA \mod 1)]$ for $A = (\sqrt{5} - 1)/2$. Compute the locations to which the keys 61, 62, 63, 64 and 65 are mapped.

2. **Exercise 11.4-1 (p. 244) (25 points)**

   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 $h_2(k) = 1 + (k \mod (m - 1))$.

3. **Exercise 12.1-3 (p. 256) (25 points)**

   Give a non-recursive algorithm that performs an inorder tree walk. (Hint: There is an easy solution that uses a stack as an auxiliary data structure and a more complicated but elegant solution that uses no stack but assumes that two pointers can be tested for equality.)

4. **Exercise 12.3-3 (p. 264) (30 points)**

   We can sort a given set of $n$ numbers by first building a binary search tree containing these numbers (using TREE-INSERT repeatedly to insert the numbers one by one) and then printing the numbers by an inorder tree walk. What are the worst-case and best-case running times for this sorting algorithm?