This assignment covers textbook Chapter 11 and Chapter1~8.

1. **Hash Table** (20 points)
Exercises 11.2-1, page 261

2. **Hash Function** (60 points, 15 for each sub-question)
Consider inserting keys 3,4,2,5,1 in the order given into a hash table of length m = 5 using hash function $h(k) = k^2 \bmod m$ (k^2 is the auxiliary function).

(1) Using $h(k)$ as the hash function, illustrate the result of inserting these keys using chaining. Also, compute the load factor α for the hash table resulting from the insertions.

(2) Using $h(k)$ as the primary hash function, illustrate the result of inserting these keys using open addressing with linear probing.

(3) Using $h(k)$ as the primary hash function, illustrate the result of inserting these keys using open addressing with quadratic probing, where $c1$=1 and $c2$=2.

(4) What different values can the hash function $h(k) = k^2 \bmod m$ produce when m = 11? Carefully justify your answer in detail.

3. **Algorithm Design** (20 points)
Consider an unsorted array A of n integers, design an *efficient* algorithm to determine if the array contains two integers such that they add up to a specific target number s. That is: if we can find A[i] + A[j] == s ($1 \le i, j \le n$, $i \ne j$), s is an integer, the algorithm should return TRUE, otherwise return FALSE.

Design requirement: the *efficient* algorithm you are going to design should provide a linear running time, rather than a $O(n^2)$ running-time brute-force solution or a O(nlgn) solution. You may use the algorithms that we learned in the textbook.

(1) Write the Pseudo-code *(please use textbook conventions)* (15 points)
(2) Justify the running time of the algorithm (5 points).