**2.** **[Tests Java language and prob solving]** Write a Java method testForSum which determines whether a given array of integers contains three entries whose sum is equal to a given integer. Assume the following:

a. The method accepts an array intArr of int’s and an int testInt as its two arguments

b. The return type of the method is boolean

c. The method returns true if and only if there are distinct integers i, j, k such that intArr[i] + intArr[j] + intArr[k] equals testInt.

Test your method in a main method, which passes the following input values  
{5, 1, 23, 21, 17, 2, 3, 9, 12}, 22  
into the method testForSum, and which outputs the return value to the console.

* **Algorithm:**
  1. Given an array of length *n* and a sum *s*
  2. Create three nested loop first loop runs from start to end (loop counter i), second loop runs from i+1 to end (loop counter j) and third loop runs from j+1 to end (loop counter k)
  3. The counter of these loops represents the index of 3 elements of the triplets.
  4. Find the sum of ith, jth and kth element. If the sum is equal to given sum. Print the triplet and break.
  5. If there is no triplet, then print that no triplet exist.

class FindTriplet {

    // returns true if there is triplet with sum equal

    // to 'sum' present in A[]. Also, prints the triplet

    boolean find3Numbers(int A[], int arr\_size, int sum)

    {

        int l, r;

        // Fix the first element as A[i]

        for (int i = 0; i < arr\_size - 2; i++) {

            // Fix the second element as A[j]

            for (int j = i + 1; j < arr\_size - 1; j++) {

                // Now look for the third number

                for (int k = j + 1; k < arr\_size; k++) {

                    if (A[i] + A[j] + A[k] == sum) {

                        return true;

                    }

                }

            }

        }

        // If we reach here, then no triplet was found

        return false;

    }

/\* Driver program to test above function \*/

int main()

{

    int A[] = { 1, 4, 45, 6, 10, 8 };

    int sum = 22;

    int arr\_size = sizeof(A) / sizeof(A[0]);

    find3Numbers(A, arr\_size, sum);

    return 0;

}

**3.** **[Tests knowledge of data structures]** Create your own linked list (do not use any of the classes provided in the Collections API). Implement the following two operations:

If you are using jdk1.4 or before:

void add(Object ob);

boolean find(Object ob);

String toString();

If you are using j2se5.0 and you know generic programming:

void add(T ob);

boolean find(T ob);

String toString()

The toString method should arrange the elements of the list in a comma-separated sequence, in the following format:

[elem0, elem1, elem2, …, elemN]

Test your linked list in a main method which does the following:

a. Creates an instance of your list and adds the following Strings to it:  
“Straight”, “Bent”, “Equals”, “Well”, “Storm”

b. Uses your find function to search for the keys “Well” and “Strength”

c. Outputs both the input list and the search results to the consoleand output the results to the consoleby repeatedly using your add function to populate a new instance of your linked list with Strings, and then outputting to console the boolean result of searching for some String in this list.

**4. [Tests basic knowledge of recursion]** Write a recursive static Java method that accepts an array arr of integers argument returns a list of all permutations of these integers.

(A permutation of a sequence of integers is a re-arrangement of the integers. For example, one permutation of  1, 3, 4, 8, 2 is 3, 1, 2, 8, 4.) For this problem, you may assume that the input array contains no duplicate entries. Your method should return an ArrayList of int arrays.

Next, test your method using a main method; the main method should pass in the following array: [1, 5, 4, 2]; then, it should print to the console the resulting list of permutations.

**5. [Tests knowledge of concept of static]** Create a Java class that allows at most 5 instances of itself to be created. Call your class JustFive. Provide a main method in your class that attempts to create 6 instances of your class.