Problem 6.9

```
public class q69 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		System.out.println(
				"\nFeet     Meters     |     Meters     Feet\n" +
				"-------------------------------------------");

				for (double feet = 1.0, meters = 20.0; feet <= 10.0; feet++, meters
+= 5) {
			System.out.printf("%4.1f     ", feet);
			System.out.printf("%6.3f", footToMeter(feet));
			System.out.print("     |     ");
			System.out.printf("%-11.1f", meters);
			System.out.printf("%7.3f\n", meterToFoot(meters));

				}

	}

	/** Convert from feet to meters */
	public static double footToMeter(double foot) {
		return 0.305 * foot;
	}

	/** Convert from meters to feet */
	public static double meterToFoot(double meter) {
		return 3.279 * meter;

	}

}
```

Problem 6.12

```
public class q612 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

				final int NUMBER_OF_CHARS_PER_LINE = 10;
				char ch1 = '1'; // Start
				char ch2 = 'Z'; // End

				System.out.println("\nCharacters per 1 to Z");
				printChars(ch1, ch2, NUMBER_OF_CHARS_PER_LINE);
```

```
                        System.out.println();
                }

                /** Method printChars: Prints characters ch1 and ch2 with the specified
                    numbers per line.                                          */
                public static void printChars(char ch1, char ch2, int
                numberPerLine) {
                        for (char c = ch1, count = 1; c <= ch2; c++, count++) {
                                if (count % numberPerLine == 0)
                                        System.out.println(c);
                                else
                                        System.out.print(c + " ");
                        }
                }
        }
```

Problem 6.16

```
public class q616 {
        public static void main(String[] args) {

      for (int i = 2000; i <= 2020; i++) {

        System.out.println("Year: " + i + " = " + numberOfDaysInAYear(i) + "days: is Leap year? "
+ isLeapYear(i));
      }
   }

   public static int numberOfDaysInAYear(int year) {

      if (isLeapYear(year)) return 365;
      else return 366;
   }
   public static boolean isLeapYear(int year) {

      return  (year % 4 == 0 && year % 100 != 0) || year % 400 == 0;

   }
}
```

Problem 6.18

```
import java.util.Scanner;

public class q618 {

        private static final int PASSWORD_REQUIRED_LENGTH = 0;
```

```
        public static void main(String[] args) {
                // TODO Auto-generated method stub

                Scanner input = new Scanner(System.in);
    System.out.print(
            “- A password must have at least eight characters.\n" +
            “- A password consists of only letters and digits.\n" +
            “- A password must contain at least two digits \n" +
            "Enter a password meeting the requirements above: ");
    String s = input.nextLine();

    if (isValidPassword(s)) {
        System.out.println("Password is VALID: " + s);
    } else {
        System.out.println("NOT VALID PASSWORD: " + s);
    }

}

public static boolean isValidPassword(String password) {

    if (password.length() < PASSWORD_REQUIRED_LENGTH) return false;

    int charCount = 0;
    int numCount = 0;
    for (int i = 0; i < password.length(); i++) {

        char ch = password.charAt(i);

        if (isNumeric(ch)) numCount++;
        else if (isLetter(ch)) charCount++;
        else return false;
    }


    return (charCount >= 2 && numCount >= 2);
}

public static boolean isLetter(char ch) {
    ch = Character.toUpperCase(ch);
    return (ch >= 'A' && ch <= 'Z');
}


public static boolean isNumeric(char ch) {

    return (ch >= '0' && ch <= '9');
}
```

```
}
```

Problem 7.2

```
import java.util.Scanner;

public class q72 {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		int[] numbers = new int[10];
		java.util.Scanner input = new java.util.Scanner(System.in);

		System.out.print("EnterNumers: ");
		for(int i=0; i<10; i++) {
			numbers[i] = input.nextInt();

		}
		for(int i=9; i>=0; i--) {
			System.out.println(numbers[i]);
		}

	}

}
```

Problem 7.4

```
import java.util.Scanner;

public class q74 {

	static final int MAX = 100;
	public static void main(String[] args) {

	   int[] scores = new int[MAX];
	   int numberOfScores = 0;

	   Scanner input = new Scanner(System.in);

	   System.out.print("Enter grades: ");
	   for (int i = 0; i < MAX; i++) {

	      int num = input.nextInt();

	      if (num < 0)  break;

	      scores[i] = num;
```

```
            numberOfScores++;
        }
        scores[numberOfScores] = -1;
        int average = getAverage(scores, numberOfScores);
        int aboveAETA = scoresAboveAndEqualToAverage(scores, average);

        System.out.println("Number of scores: " + numberOfScores);
        System.out.println("Average score is: " + average);
        System.out.println("Scores above average = " + aboveAETA);
        System.out.println("Scores below average = " + (numberOfScores - aboveAETA));

    }

    public static int getAverage(int[] scores, int numberOfScores) {
        int total = 0;
        for (int i = 0; scores[i] >= 0; i++) {
            total += scores[i];
        }

        return total / numberOfScores;
    }

    public static int scoresAboveAndEqualToAverage(int[] scores, int average) {
        int count = 0;
        for (int i = 0; scores[i] >= 0; i++) {

            if (scores[i] >= average) count++;
        }

        return count;
    }
}
```

Problem 7.8

```
import java.util.Scanner;
public class q78 {

    public static void main(String[] args) {
            // TODO Auto-generated method stub
            final int SIZE = 10;

        double[] numbers = new double[SIZE];
        Scanner input = new Scanner(System.in);
        System.out.print("Enter 10 double numbers: ");
        for (int i = 0; i < numbers.length; i++) numbers[i] = input.nextDouble();

        System.out.println("The average value is: " + average(numbers));
```

```
    }

    public static int average(int[] array) {
        int total = 0;
        for (int i = 0; i < array.length; i++) {
            total += array[i];
        }
        return total / array.length;
    }

    public static double average(double[] array) {
        double total = 0;
        for (int i = 0; i < array.length; i++) {
            total += array[i];
        }
        return total / array.length;
    }

    public static void printArray(int[] array, int numberPerLine) {

        for (int i = 0; i < array.length; i++) {

            System.out.printf("%4d", array[i]);
            if ((i + 1) % numberPerLine == 0) System.out.println("");
        }
    }
}
```