1.
```
import chn.util.*;
public class Problem1
{
    // $5/1 gallon * 1 gallon/30 miles * 45 miles/1 hr * 1 hr/60 min * m min
    // = $m/8
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("m = ");
        double m = keyboard.readDouble();
        // round to nearest hundredth
        // multiply by 100, round to nearest int, then divide by 100
        // e.g.: 74.5098039... --> 7450.98039... --> 7451 --> 74.51
        System.out.println("$" + (double)Math.round(m / 8 * 100)/100);
    }
}
```

2.
```
import chn.util.*;
public class Problem2
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("x1 = ");
        double x1 = keyboard.readDouble();
        System.out.print("y1 = ");
        double y1 = keyboard.readDouble();
        System.out.print("x2 = ");
        double x2 = keyboard.readDouble();
        System.out.print("y2 = ");
        double y2 = keyboard.readDouble();
        if(x1 == x2) {
            System.out.println("(a) undefined");
        }
        else {
            System.out.println("(a) " + (y2-y1)/(x2-x1));
        }
        System.out.println("(b) (" + (x1+x2)/2 + ", " + (y1+y2)/2 + ")");
        System.out.println("(c) " + Math.sqrt(Math.pow(x1-x2,2)+Math.pow(y1-y2,2)));
    }
}
```

3.
```
import chn.util.*;
public class Problem3
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("a = ");
        int a = keyboard.readInt();
        System.out.print("b = ");
        int b = keyboard.readInt();
        int sum = 0;
        for(int i=a; i<=b; i++) {
            if(i % 2 == 0) { // even
                sum += Math.pow(i, 2);
            }
            else { // odd
                sum += Math.pow(i, 3);
            }
        }
        System.out.println(sum);
    }
}
```

4.
```
import chn.util.*;
public class Problem4
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("Sring: ");
        String str = keyboard.readLine();
        System.out.print("n = ");
        int n = keyboard.readInt();
        String newString = "";
        int letterVal;
        for(int i = 0; i < str.length(); i++) {
            if(isLetter(str.charAt(i))) {
                letterVal = str.toLowerCase().charAt(i) - 'a' + 1;
                if(letterVal % n != 0) {
                    newString += str.charAt(i);
                }
            }
            else {
                newString += str.charAt(i);
            }
        }
        System.out.println(newString);
    }
    public static boolean isLetter(char c) {
        return ((c >= 'a') && (c <= 'z')) || ((c >= 'A') && (c <= 'Z'));
    }
}
```

5.
```
import chn.util.*;
```

```
public class Problem5
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("a = ");
        int a = keyboard.readInt();
        System.out.print("Word: ");
        String w = keyboard.readLine();
        int result = 0;
        int letterVal;
        for(int i=0; i<w.length(); i++) {
            letterVal = w.toLowerCase().charAt(i) - 'a' + 1;
            result += factorial(i+1) * (letterVal - a);
        }
        System.out.println(result);
    }
    // factorial:  recursive function to compute n!
    // uses the recursive definition of factorial n! = n * (n-1)!
    public static int factorial(int n) {
        // base case:  0! = 1
        if(n == 0)
            return 1;
        // recursive case:  n! = n * (n-1)!
        else
            return n * factorial(n-1);
    }
}
```

6.
```
import chn.util.*;
public class Problem6
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("a = ");
        int a = keyboard.readInt();
        System.out.print("b = ");
        int b = keyboard.readInt();
        System.out.print("n = ");
        int n = keyboard.readInt();
        int num = 0;
        for(int i=a; i<=b; i++) {
            if(sumOfDigits(i) % n == 0) num++;
        }
        System.out.println(num);
    }
    // sumOfDigits: compute sum of digits of number x
    public static int sumOfDigits(int x) {
        String s = "" + x + ""; // convert number into string
        int sum = 0;
        for(int i=0; i<s.length(); i++) {
            sum += s.charAt(i) - '0'; // extract each digit from string
        }
        return sum;
    }
}
```

7.
```
import chn.util.*;
public class Problem7
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("a = ");
        int a = keyboard.readInt();
        System.out.print("b = ");
        int b = keyboard.readInt();
        int sum = 0;
        for(int i=a; i<=b; i++) {
            if(isPalindrome(i)) sum += sumOfDigits(i);
        }
        System.out.println(sum);
    }
    // isPalindome: return whether number x is palindrome
    public static boolean isPalindrome(int x) {
        String s = "" + x + ""; // convert number into string
        String reversed = "";
        for(int i=s.length()-1; i>=0; i--) {
            // reverse string by appending characters starting from back of string
            reversed = reversed + s.charAt(i);
        }
        return s.equals(reversed);
    }
    // sumOfDigits: compute sum of digits of number x (see #6 solution)
    public static int sumOfDigits(int x) {
        String s = "" + x + ""; // convert number into string
        int sum = 0;
        for(int i=0; i<s.length(); i++) {
            sum += s.charAt(i) - '0'; // extract each digit from string
        }
        return sum;
    }
}
```

8.
```
import chn.util.*;
```

```
public class Problem8
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("Word 1: ");
        String w1 = keyboard.readLine();
        System.out.print("Word 2: ");
        String w2 = keyboard.readLine();
        // compute averages
        double avg1 = 0, avg2 = 0;
        for(int i = 0; i < w1.length(); i++) {
            avg1 += letterValue(w1, i);
            avg2 += letterValue(w2, i);
        }
        avg1 /= w1.length();
        avg2 /= w1.length();
        // iterate through strings again
        double cov = 0;
        for(int i = 0; i < w1.length(); i++) {
            cov += (letterValue(w1, i) - avg1) * (letterValue(w2, i) - avg2);
        }
        System.out.println(cov);
    }
    // returns numerical value of letter at specific index in string
    public static int letterValue(String w, int index) {
        return w.toLowerCase().charAt(index) - 'a' + 1;
    }
}
```

9.
```
import chn.util.*;
public class Problem9
{
    public static void main(String[] args) {
        ConsoleIO keyboard = new ConsoleIO();
        System.out.print("n = ");
        int n = keyboard.readInt();
        // first, determine primality of integers 2 to 9999
        // using Sieve of Eratosthenes:
        boolean[] isPrime = new boolean[10000];
        // first, assume all are prime
        for(int i=2; i<=9999; i++)
            isPrime[i] = true;
        // iterate through candidates for start number: 2 to 99
        for(int i=2; i<=99; i++) {
            // if this integer is prime, then all its multiples must be marked not prime
            if(isPrime[i]) {
                // mark 2i, 3i,... as not prime
                for(int j=2*i; j<=9999; j+=i)
                    isPrime[j] = false;
            }
        }
        // add up primes
        int sum = 0;
        for(int i=2; i<=n; i++) {
            if(isPrime[i]) sum += i;
        }
        System.out.println(sum);
    }
}
```