

A**Problem 1**

Using only pen and paper find how we should fill in the following code in the dotted spaces so that all `println` statements print four times the same value:

```
int n1d = 23;
int n1b = 0b...
int n1o = 0.....
int n1x = 0x...
System.out.println(n1d + " " + n1b + " " +
                    n1o + " " + n1x);

int n2d = 188;
int n2b = 0b...
int n2o = 0.....
int n2x = 0x...
System.out.println(n2d + " " + n2b + " " +
                    n2o + " " + n2x);

int n3d = 599;
int n3b = 0b...
int n3o = 0.....
int n3x = 0x...
System.out.println(n3d + " " + n3b + " " +
                    n3o + " " + n3x);
```

B

How to create a number of type `int` so its bit representation has 1 at the k th position (positions are counted from zero for the least significant bit) and 0 at all other positions. Create a few such numbers for different values of k and print them.

C

Define a number of type `int` and using ANDing operation for a few values of k check whether the bit at the position k in this number is 0 or 1.

D

Given an integer n , define a number k whose bit representation has all zeros but the least significant byte is identical to the least significant byte of n . For example, if n is -1 , the value of k should be 255 (why?).

E

Given an integer n , define a number k whose bit representation has all zeros but the least significant byte is identical to the second least significant byte of n . For example, if n is 1793 , the value of k should be 7 (why?).

F

Given a number n , define a number k which has bit representation identical to that of n but with the bit at position 5 flipped (positions are counted from 0 for the least significant bit). For example, if n is 65, then

```
System.out.println((char)n + " " + (char)k);
```

should print A a, and if n is 70, we should see F f (why?). Hint: use XORing.

G

Given two positive integers n and m , both smaller than 256, define a number k with the second to least significant byte identical to the least significant byte of n and the least significant byte identical to the least significant byte of m . For example, if n is 3 and m is 15, then k should be 783.

H

Given a number n define a number k whose bit representation is identical to that of n , but with all bits of the second least significant byte flipped. For example, if n is 65025, then k should be 257.

I

Using only pen and paper, find what will be printed by the following fragment of code:

```
byte b = 0x7F;  
System.out.print( (byte)(b << 1) + " ");  
System.out.print( (byte)(b << 3) + " ");  
System.out.print( (byte)(b << 5) + " ");  
System.out.print( (byte)(b << 7) + " ");  
System.out.print( (byte)(b << 8) + "\n");
```

What would be the result without casting on `byte` the outcomes of bit-shifting?

Problem 2

Using only pen and paper, calculate what the following program will print

```
int a = 37, b = 98;  
System.out.println("a & b          " + (a & b));  
System.out.println("a | b          " + (a | b));  
System.out.println("a ^ b          " + (a ^ b));  
System.out.println("a << 2         " + (a << 2));  
System.out.println("(a >> 1) & b      " + ((a >> 1) & b));  
System.out.println("(a << 2) ^ (b >> 2) " + ((a << 2) ^ (b >> 2)));  
System.out.println("a & ~(b >> 3)   " + (a & ~(b >> 3)));
```

[download BitOp.java](#)

and then check the results by running the program.

Problem 3

Write a program defining four natural numbers from the range [0, 255] and representing components of a color: α , red, green, blue. Then pack these four numbers into

one **int**, bytes of which (from the most to the least significant) correspond to the four components of the color in the order given above.

Using only this integer, unpack it to four numbers corresponding to the four components of the color.

For example, the following program

```
public class ColorBytes {
    public static void main(String[] args) {
        int a = 23, r = 121, g = 255, b = 130, color = 0;

        // pack four components into one int color
        //
        // ...
        //

        System.out.println("a, r, g, b = " + a +
                           ", " + r + ", " + g + ", " + b);
        System.out.println("color = " + color);
        a = r = g = b = 0;

        // unpack color to get its four
        // components back in a, r, g, b
        //
        // ...
        //

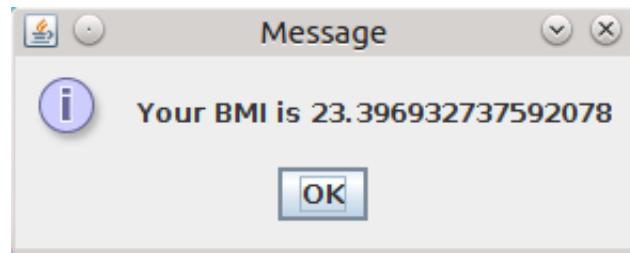
        System.out.println("a, r, g, b = " + a +
                           ", " + r + ", " + g + ", " + b);
    }
}
```

should print

```
a, r, g, b = 23, 121, 255, 130
color = 393871234
a, r, g, b = 23, 121, 255, 130
```

Problem 4

Write a program which asks the user, by means of an object of type **Scanner**, to enter his/her height (in meters, as **double**), then the weight (in kilograms, also as a **double**). Then the program displays in a message box (**JOptionPane.showMessageDialog**) his/her BMI coefficient (*body mass index*) defined as the weight in kilograms divided by the square of height in meters — this number should come out close to 20.



You can also just show the result on the console.
