

Task 5

A date may be specified by three numbers denoting year, month and day of month. If we don't need dates from a deep past, we can specify years relative to the year 2000 (e.g., year 2032 as 32). Assuming that we will not consider the future past the year 2127, we can use numbers from the interval $[0, 127]$ for year; such numbers can be written on 7 bits. Month number can occupy 4 bits, as it belongs to the interval $[0, 15]$, and for the day we can spend 5 bits (which is enough to store any number from the interval $[0, 31]$). Therefore, full date can be packed in 16 bits and we can store two such dates in one 32-bit `int` (e.g., beginning and end of a period).

Write a program which defines two dates (*from* and *to*), each as three integers (year, month, day). Pack them into one integer. Then unpack the integer and recover these six numbers.

For example, the following program

[download DatesSimple.java](#)

```
public class DatesSimple {
    public static void main (String[] args) {
        int fromy = 2000, fromm = 2, fromd = 3;
        int  toy = 2127,  tom = 11,  tod = 29;

        System.out.println("**** Original\n" +
                           "From " + fromy + "/" +
                           fromm + "/" + fromd + " to " +
                           toy + "/" + tom + "/" + tod);

        int period = 0;

        // ... (pack 6 numbers into 'period')

        fromy = fromm = fromd = toy = tom = tod = 0;

        // ... (unpack 6 numbers from 'period')

        System.out.println("**** Reconstructed\n" +
                           "From " + fromy + "/" +
                           fromm + "/" + fromd + " to " +
                           toy + "/" + tom + "/" + tod);
    }
}
```

should print

```
**** Original
```

From 2000/2/3 to 2127/11/29

**** Reconstructed

From 2000/2/3 to 2127/11/29

Deadline: Apr 30 (inclusive)

Put your Java file(s), and only Java files, in a directory the name of which is your surname (without Polish or any other non-ASCII characters). Names of Java files are arbitrary, although of course they should correspond to names of classes you created. Zip the whole directory ("from above" — not just the files inside it). Then drop the zip file created in this way into folder "Tasks / Task_XX" of the GAKKO system (where 'XX' is the task number).