

```

/*--- HW14.java ---*/
import java.util.*;

/**
 * Driver for HW14
 * adapted from given source code <a href="http://faculty.cs.usna.edu/IC211/calenda
r.php?key=d23572f0088fdc0821b77dd414d4ca4b22f7d51a&type=class&event=14">
 * @author Hanling, Mike - 202430
 */
public class HW14 {
    /**
     * The main function from HW14:
     * Determine whether to work in verbose mode or not
     * Output instructions if in verbose mode
     * Read in three ints
     * Save the value from doing computation on those three ints in MyMath.modexp
     * Print the result back to the terminal
     * Error checking is done for read in of ints, and value checking for the
     * computation
     * @param -v Optional - will put in verbose mode
     */
    public static void main(String[] args) {
        boolean verbose = args.length > 0 && args[0].equals("-v");
        Scanner sc = new Scanner(System.in);

        if (verbose) {
            System.out.print("Enter x, k, m: ");
        }
        Integer x = null;
        Integer k = null;
        Integer m = null;
        Integer r = null;
        try {
            x = sc.nextInt();
            k = sc.nextInt();
            m = sc.nextInt();
            r = MyMath.modexp(x, k, m);
        } catch (Throwable e) {
            System.out.print( (verbose ? "Error in HW14! invalid input\n" : ""));
            System.exit(1);
        }

        if (verbose) {
            System.out.print(x + "^" + k + " % " + m + " = ");
        }
        System.out.println(r);
    }
}

/*--- MyMath.java ---*/
/**
 * Currently, this cclass holds one function for computation
 * Static method, no fields
 * adapted from given source code <a href="http://faculty.cs.usna.edu/IC211/calenda
r.php?key=d23572f0088fdc0821b77dd414d4ca4b22f7d51a&type=class&event=14">
 * @author Hanling, Mike - 202430
 */
public class MyMath {
    /**
     * Returns x^k mod m
     * Note: k must be non-negative, and m must be positive
     * @param x Int for the base of the expression
     * @param k Int for the exponent (must be non-neg)
     * @param m Int for the modulo (must be positive)
     * @return r Int answer to x^k % m
     * @throws Throwable for divide by zero
     * @throws Throwable if the three inputs do not meet specification
     */

```

```

    public static int modexp(int x, int k, int m) throws Throwable {
        if (k < 0 || m < 1)
            throw new Throwable();

        int r = 1;

        for (int i = 0; i < k; i++) {
            r = r * x % m;
        }
        return r;
    }
}

```