On the second class we learned a number of multiplication algorithms including the russian method known as “multiplication à la russe”. Find below a recursive definition of this algorithm (the comments should help you understand why it works):

\[
\text{Mulàlarusse}(a, b) \\
\text{If } (b = 0) \text{ Then return } 0 \quad // a \cdot 0 = 0 \\
\text{Else if } (b \text{ is even}) \text{ Then return } \text{Mulàlarusse}(a + a, b/2) \quad // a \cdot b = (2 \cdot a) \cdot (b/2) \\
\text{Else return } (a + \text{Mulàlarusse}(a + a, (b-1)/2)) \quad // a \cdot b = a + (2 \cdot a) \cdot ((b-1)/2)
\]

[25%]  
• 1) Write an iterative definition of this algorithm (no recursion allowed).

Now notice that if we imbed \( (\text{mod } n) \) operators in this definition we end up with an algorithm computing \( (a \cdot b) \text{ mod } n \) instead:

\[
\text{MulMOD}(a, b, n) \\
\text{If } (b = 0) \text{ Then return } 0 \quad // a \cdot 0 = 0 \text{ mod } n \\
\text{Else if } (b \text{ is even}) \text{ Then return } \text{MulMOD}((a + a) \mod n, b/2, n) \quad // a \cdot b = (2 \cdot a) \cdot (b/2) \mod n \\
\text{Else return } (a + \text{MulMOD}((a + a) \mod n, (b-1)/2, n) \mod n) \mod n \quad // a \cdot b = a + (2 \cdot a) \cdot ((b-1)/2) \mod n
\]

[30%]  
• 2) Find the largest number \( n \) of type \text{long} for which the method \text{MulMod}(a, b, n) finds the correct product \( \text{mod } n \) for all \( a, b \) with \( 0 \leq a, b \leq n-1 \) and compare it to the direct \text{Java} expression \((a \cdot b) \% n\). Write a \text{Java} program to find the largest correct values for both methods. \text{NOTE}: if you get a negative result then your answer is wrong.

[15%]  
• 3) Find a wiser way to compute \( (a+b) \text{ mod } n \) allowing you to reach larger values in \text{MulMod}(a, b, n) of part •2).
Now notice that if we replace “+” by “*” in the MulMOD algorithm and “0” by “1” in the base case, we end up defining an algorithm for $a^b \mod n$ instead of $(a^b) \mod n$:

\[
\text{ExpMOD}(a,b,n)
\]

\[
\begin{align*}
\text{If } (b=0) & \quad \text{return } 1 \quad // a^0 = 1 \mod n \\
\text{Else If } (b \text{ is even}) & \quad \text{return } \text{ExpMOD}(a^2 \mod n, b/2, n) \quad // a^b = (a^2)^{b/2} \mod n \\
\text{Else return } (a \cdot \text{ExpMOD}(a^2 \mod n, (b-1)/2, n) \mod n) \quad // a^b = a \cdot (a^2)^{(b-1)/2} \mod n
\end{align*}
\]

[30%]

•4) Write a Java program to compute $a^b \mod n$ for $a,b,n$ of type long (with $0 \leq a,b \leq n-1$)

-- REMARK: Do not forget what you did in •2) -- and run it with the values

\[
\begin{align*}
a & := 1274434408442 \\
b & := 589394265617 \\
n & := 1606818609763
\end{align*}
\]

• What is special about the answer $c := a^b \mod n$ you get?

(if your answer is correct you should notice something special…)

Let $d = 433371342353$. Run your program with the values $c,d,n$.

• What is special about the answer $e := c^d \mod n$ you get?

(if your answer is correct you should notice something…)

NOTE: For questions •2) and •4) you may implement either a recursive or iterative method.