Exam Question 3.

1. The JVM specifies that the `int` type is “32-bit signed twos complement integers.” Let’s call such an integer a “JVM int.” Define the ACL2 function `JVM-intp` that takes an ACL2 object, n, and returns `t` or `nil` according to whether n is a JVM int.

2. According to the JVM spec, “the built-in integer operators do not indicate (positive or negative) overflow in any way; they wrap around on overflow.” Suppose i and j are both JVM ints. Define the ACL2 functions `jvm+-`, `jvm--`, and `jvm-*` so that `(jvm+- i j)`, `(jvm-- i j)`, and `(jvm-* i j)` are the JVM ints obtained, respectively, by adding, subtracting, and multiplying i and j according to the JVM spec.
3. Define the M1 instructions IADD, ISUB, and IMUL by exhibiting the definitions of execute-IADD, execute-ISUB, and execute-IMUL.

Format:
(IADD)
(ISUB)
(IMUL)

Operand Stack:
..., value1, value2 => ..., result

Description:
Both value1 and value2 must be of type JVM int. The values are popped from the operand stack. The JVM int result is, respectively, value1+value2, value1-value2, or value1*value2. Result is pushed onto the operand stack.

4. Define two recursive versions of the factorial function for M1.
FACT uses the original M1 integer operations such as (MUL) and (SUB). JVM-FACT uses the JVM int operations such as (IMUL) and (ISUB).

What is the largest input value of n for which FACT returns the same integer as JVM-FACT?

Hint: If you know Java, you can check your implementation of the JVM int operations by comparing the results produced by JVM-FACT with the results produced by the following Java method for computing factorial.

```java
public static int fact(int n){
    if (n>0)
        {return n*fact(n-1);}
    else return 1;
}
```