Problem 3: Suppose I need to compute the series $f_n = f_{n-1}^2$. If the value $f_0 = 2$, what is the maximum n that can be stored in the following C++ data types, assuming that an int is 2 bytes, a long int is 4 bytes, and each byte stores 4 bits?

Let us try out the first few terms of this recursive formula first: $f_1 = f_0^2 = 2^2$; $f_2 = f_1^2 = (2^2)^2 = 2^4 = 2^{2^2}$; $f_3 = f_2^2 = (2^4)^2 = 2^8 = 2^{2^3}$; $f_4 = f_3^2 = (2^8)^2 = 2^{16} = 2^{2^4}$.
So we can see that this function f_n can be rewritten as

$$f_n = 2^{2^n}. \tag{1}$$

The next thing we should notice about is that in this question each byte can only stores 4 bits instead of 8.

a) int
An int is 2 bytes which is equal to 8 bits, so the size of it is 2^8. However since int contains negative numbers, so the maximum of int is

$$\frac{2^8}{2} - 1 = 2^7 - 1, \tag{2}$$

where the -1 is the counting for 0. Since $f_2 < 2^7 - 1 < f_3$, the maximum n can be stored in an int is 2.

b) long int
A long int is 4 bytes which is equal to 16 bits, so the size of it is 2^{16}. Again half the size of it is used to represent negative numbers, so the maximum of long int is

$$\frac{2^{16}}{2} - 1 = 2^{15} - 1, \tag{3}$$

Since $f_3 < 2^{15} - 1 < f_4$, the maximum n can be stored in a long int is 3.

c) unsigned long int
This time the long int is unsigned, so the range of it is from 0 to $2^{16} - 1$. Since $f_3 < 2^{16} - 1 < f_4$, the maximum n can be stored in an unsigned long int is still 3.