1. What is the 32 bit sequence best representing the number 1/3 in floating point? Explain exactly how you derive this. Check your result using the program showbin.f. What is the relative error of the approximation to 1/3 (give your answer as an ordinary decimal number to two significant digits)? How does this compare to the largest possible relative error in the 32 bit floating point number system?

2. We computed \( g_n = \int_0^1 x^n e^{-x} \, dx \) for \( n = 20 \) with the unstable recursion
\[
g_0 = 1 - e^{-1}, \quad g_n = 1 - ng_{n-1}, \quad n = 1, 2, \ldots, 20,
\]
and got very bad results. Consider instead rearranging the formula to obtain the backward recursion
\[
g_{n-1} = \frac{(1 - g_n)}{n}, \quad n = 40, 39, \ldots, 21.
\]
The problem is that we need to start with \( g_{40} \) which, of course, we do not know. Suppose we choose \( g_{40} = 0 \) (which is obviously wrong). Assuming that we carry out the recursion exactly, what is the absolute and relative error in our computed value for \( g_{20} \)? Is the computed value for \( g_{35} \) correct to at least 6 significant digits? Check this out both theoretically and on the computer and report your results.