PRF Theory

PRF can be defined as follows: PRF is a pseudorandom function which should be easy to compute and should be difficult to distinguish between a random function mapping and PRF mapping of any input.
$Pr[D(F(s)) = 1] - Pr[D(f(s)) = 1] <= negl(n), where\ s \in \{0, 1\}^n$

This means that a distinguisher D wouldn't be able to distinguish between a real random function and a PRF with a probability not greater than negligible.

The task here is to convert the PRG generated to PRF. $F_k(s)$ is to be computed, which returns a n length string given input to be a $2n$ length string.
The procedure for each generated $G(s)$ which is a $2n$ length string is to divide into two sections $G_0(s)$ and $G_1(s)$, and one of them is chosen as the seed for the next generator by looking at the corresponding position in the key, i.e.,

$$F_k(x_1x_2x_3x_4...x_n) = G_{x_n}(...(G_{x_2}(G_{x_1}(k))))$$

Similar to the first question, since the discrete log problem is hard, the modular exponentiation is one way with any seed. Counting on the construction of the pseudorandom generator, the PRF algorithm will result in pseudo-random n length strings.