Description:

1. Finding Prime Implicants:
   The `find_prime_implicants` function iteratively combines terms differing by one bit to generate prime implicants. It:
   - Groups terms based on 1's.
   - Combines adjacent groups, replacing differing bits with a '-' to indicate generalization.
   - Tracks terms that cannot be further combined as prime implicants.
2. Covering Minterms:
   - `covers` Function: Checks if a prime implicant covers a specific minterm.
   - `generatePrimeImplicantTable` Function: Constructs a table mapping prime implicants to the minterms they cover.
3. Finding Essential Prime Implicants:
   The `findEssentialPrimeImplicants` function identifies essential prime implicants—those that uniquely cover a minterm. It iterates through minterms, selects unique prime implicants, and ensures all minterms are covered.
4. Process Flow:
   - The input includes minterms and don't-cares (binary strings).
   - Prime implicants are derived through grouping and reduction.
   - Essential prime implicants are identified to minimize the Boolean function.

Challenges:

I had some challenges getting the most minimal output as the input size grew larger. I was able to get the solution to within a few terms of the fully minimized solution, but not all of the way. It does work really well with smaller inputs though.