Problem 3.1

A) Consider the following proposition:
- Let $L(f)$ be the number of literals for function f. The number of NAND2 gates required to implement f is $L(f) - 1$.
  - Prove this proposition assuming f is a two-level logic.
  - Prove this proposition assuming f is a multi-level logic.

B) Prove the propositions 3 and 4 on the cube-literal matrix described in the slide "Cube-Literal Matrix (3)"

C) Compute all kernels for the function

$$F = abde + cde + bcdf + aef$$

Problem 3.2

A) Consider the below three functions :

$F = abde + cde + bcd + ace$

$G = abd + bce$

$H = abe + acd$

i. Construct the cube-literal matrix

ii. Identify all common cubes (rectangle with |C| ≥ 2 and |R| ≥ 2) in the matrix

iii. For each extracted common cubes, compute the # of gates saved when algebraic division is applied.

iv. Select the common cube with the largest gate savings and apply algebraic division on the corresponding functions.

v. Continue the process of iii and iv until no gate savings is possible.

Problem 3.2

B) Consider the following three functions :

$$F = \overset{1}{ad} + \overset{2}{ac} + \overset{3}{bed} + \overset{4}{bef} + \overset{5}{cf}$$

$$G = \overset{6}{ac} + \overset{7}{bce} + \overset{8}{bd} + \overset{9}{bf}$$

i. Construct the cokernel-cube matrix (use the cube indices as indicated above).

ii. Identify all non-trivial kernel intersections (rectangle with $|C| \geq 2$ and $|R| \geq 2$) in the matrix

iii. For each non-trivial kernel intersections, compute the # of gates saved when algebraic division is applied.

iv. Select the non-trivial kernel intersection with the largest gate savings and apply algebraic division on the corresponding functions.

v. Continue the process of iii and iv until no gate savings is possible.

Problem 3.3 *(extra-credit)*

Write a program which computes all kernels for a given function

i. Input function is to be given as a set of cubes in cube-literal matrix.

ii. Input functions are to be given as a set of cubes in cube-literal matrix.

iii. Display the kernels in the form of cube-literal matrices. Also, if possible, display the input function and its kernels in equation form (Each literal corresponding to the column of the cube-literal matrix could be merely labeled as a, b, c, d, …)