Problem 2.1

Consider the following function :

 $f = \bar{a}\bar{b}d + ab\bar{c} + b\bar{c}\bar{d} + a\bar{b}\bar{d} + ac + \bar{a}\bar{c}\bar{d}$

- A) Construct a 0-cube table (list of ON-set minterms in bit-vector form).
- B) Give the minimum set of prime implicants to implement *f* by Quine-McClusky method:
 - Compute all prime implicants by generating 1-cube table, 2-cube table, and so on, by finding adjacent cube pairs and reducing the cube pairs.
 - Generate the prime implicant table (columns : prime implicants, rows : minterms)
 - Compute the minimum set of prime implicants to cover all minterms
 - \checkmark Elimination of dominated primes and dominating minterms
 - \checkmark Extraction of essential primes

Problem 2.2

Consider the same function :

$$f = \bar{a}\bar{b}d + ab\bar{c} + b\bar{c}\bar{d} + a\bar{b}\bar{d} + ac + \bar{a}\bar{c}\bar{d}$$

- A) Compute the complement function \overline{f} by Shannon Expansion method
- B) Compute the complement of the function derived in A) by Negate-and-Expand method. (Confirm that the result is the same as the prime implicant set derived in Problem 2.1)

C) Generate the reduced prime table

- Partially redundant prime set at each column (eliminate essential prime set and totally redundant prime set)
- Minterms (or a collection of minterms) at each row which needs to be covered by the columns (minterms covered by the essential primes are to be eliminated)
- D) Compute the minimum set of prime implicants from the reduced prime table.

Problem 2.3 (extra credit)

Write a program which performs a two-level logic minimization according to the following specifications :

- Input logic function is to be described as a set of cubes in either the ON-set or DC-set. Each cube should be in bitvector representation on ASCII text ('0', '1', '-' for input variables; '1' or 'x' for output to denote ON-set or DC-set). You can decide the format details to your convenience.
- Output (optimized) logic function is to be described in the same way as the input.
- You can use whatever algorithm you like (including the ones not introduced in this lecture)
- ✓ The program should be able to handle 4 or more input variables and 1 or more output variable.

Program can be written in any programming language
In the report, explain the algorithm, submit the entire source code, and give the optimization results for 2 or more functions. Also add some discussions about your program (how you tried to improve the computation time, special data structure you used, new ideas in the algorithm, etc.)

-0-1	1
1-10	1
-01-	х
01	х
11	x