Practical List

1. Create a class SET. Create member functions to perform the following SET operations:
    a. is member: check whether an element belongs to the set or not and return value as true/false.
    b. powerset: list all the elements of the power set of a set.
    c. subset: check whether one set is a subset of the other or not.
    d. union and intersection of two Sets.
    e. complement: assume universal set as per the input elements from the user.
    f. set difference and symmetric difference between two sets.
    g. cartesian product of sets.

    Write a menu driven program to perform the above functions on an instance of the SET class.
2. Create a class RELATION, use Matrix notation to represent a relation. Include member functions to check if the relation is Reflexive, Symmetric, Anti-symmetric, Transitive. Using these functions check whether the given relation is: Equivalence or Partial Order relation or None
3. Write a Program that generates all the permutations of a given set of digits, with or without repetition.
4. For any number n, write a program to list all the solutions of the equation $x_1 + x_2 + x_3 + ... + x_n = C$, where C is a constant ($C<=10$) and $x_1, x_2, x_3, ..., x_n$ are nonnegative integers, using brute force strategy.
5. Write a Program to evaluate a polynomial function. (For example store $f(x) = 4n^2 + 2n + 9$ in an array and for a given value of n, say n = 5, compute the value of f(n)).
6. Write a Program to check if a given graph is a complete graph. Represent the graph using the Adjacency Matrix representation.
7. Write a Program to accept a directed graph G and compute the in-degree and out degree of each vertex.