Programming Assignment 3: Modeling the Average Aggregate Throughput for Permutation Patterns on Hypercube with dimension-order and all shortest path routing.

**Permutation:** A communication pattern can be represented as a set of source-destination pairs. For example, in a system with 4 nodes 0, 1, 2, 3. The set {0->1, 1->2, 1->3} is a communication pattern. A permutation is a pattern where each node can at most be the sender once and the receiver once. Hence, {0->1, 1->2, 1->3} is not a permutation while {0->1, 1->2, 2->3} is a permutation. Permutation is often used to evaluate the performance of interconnects. The first step in this assignment is to write a program to generate random permutations.

In the second step, you will write programs that compute the modeled throughput for a give permutation: for dimension-order routing (single paths routing), implement Methods 1, 2, 4; for all-path routing, implement Methods 3 and 5. For method 1, the program should take a permutation file as input and directly output the aggregate throughput. For Methods 2, 3, 4, and 5, the program should take a permutation file as input and output a file in .lp format (linear programming file) for CPLEX. You can then call CPLEX, installed on crux.cs.fsu.edu, to solve the linear programming formulation to obtain the aggregate throughput for a permutation. Let us assume that each link in the hypercube is 1Gbps in the modeling.

Finally, you last step is to write a driver program that would randomly produce 100 random permutations, find aggregate throughput for the 100 permutations, and report the average throughput for the 100 random permutations.