College of Computing and Information Technology

Course Code: CCS4210 **Course Name:** High Performance Computing

Dr. Hanan Hassan **TA:** Marwa Alazab

Sheet 2: Task Decomposition

1. Parallel Logical Evaluation

   Write a program using OpenMP that evaluates the logical expression

   Y = (A == B) AND (C != D) AND (E OR F)

   by assigning each comparison to a separate thread.

---

2. Parallel Sorting (Task Split)

   Implement a parallel merge sort using OpenMP tasks where each recursive call executes as a separate task.

---

3. Parallel File Compressor (Task Pipeline)

   Create a mini compressor with three stages (tasks):

   - Task 1: Read a file chunk
   - Task 2: Compress it (e.g., using RLE or Huffman placeholder logic)
   - Task 3: Write it to the output

   Use OpenMP tasks with dependencies to simulate a producer–consumer pipeline.

4. Parallel Sudoku Solver

   Use recursive task decomposition to solve Sudoku boards:

   - Each recursive branch (placing a number) is a new task.
   - Use synchronization to stop all threads when one valid solution is found.

5. Parallel Game Tree Search (Exploratory Decomposition)
   - Simulate a two-player minimax search (like Tic-Tac-Toe).
   - Each possible move spawns a new task exploring future game states.
   - Prune the search space dynamically when a winning path is found.

6. Parallel N-Queens Solver
   - Use OpenMP tasks to explore possible placements of queens recursively.
   - Each recursive call creates a task for each potential queen placement.
   - Display total valid solutions.