Instructions for Experiments E1, E2, and E3:

General Information:

- ➔ 1 clock cycle = 500 ticks in gem5.
- ➔ The cache memory files are located at:
  - ◆ gem5/src/mem/ruby/structures/CacheMemory.hh
  - ◆ gem5/src/mem/ruby/structures/CacheMemory.cc
- ➔ Use the curTick() function to track the current tick
- ➔ The L1-I cache (Instruction Cache) and L1-D cache (Data Cache) hit/miss counts can be modified in CacheMemory.hh and CacheMemory.cc.

Part A: Counting Hits and Misses

For all experiments (E1, E2, and E3), you need to modify the CacheMemory.cc (or) CacheMemory.hh files to count the hits and misses between T1 and T2.

The following functions are already defined in CacheMemory.hh:

- ➔ void profileDemandMiss();
- ➔ void profileDemandHit();
- ➔ These functions are linked to state machine files (.sm files).

In gem5/src/mem/ruby/protocol/RubySlicc_Types.sm, the functions are declared:

- ➔ void profileDemandMiss();
- ➔ void profileDemandHit();
- ➔ In gem5/src/mem/ruby/protocol/MESI_Two_Level-L1cache.sm, you can find how hits and misses are calculated. For example:
  action(uu_profileInstHit, "\uih", desc="Profile the demand hit")

Modification Tasks:

1. Track hits and misses between T1 and T2 clock cycles using curTick().
2. Modify CacheMemory.cc or CacheMemory.hh to log the hit/miss count. (Hint: You can create new functions also).
3. Ensure hits per set and hits per way are correctly recorded (for E1).
4. Track frequently accessed sets in L1-D (for E2).
5. Record eviction counts per set in L1-I and L1-D caches (for E3).

Part B: You must write some logic according to the question.