ALGORITHM FOR SET ASSOCIATIVE MAPPING

1. Start
2. Input:
   - cache_size
   - block_size
   - associativity
   - list of memory addresses to access
3. Compute number of sets:

   num_sets = cache_size / (block_size × associativity)
4. Initialize the cache as:

   cache[num_sets][associativity]

   Each block initially = None (empty)
5. For each set, initialize a FIFO replacement pointer:

   next_replace[set] = 0
6. For each memory address 'address' in the input list:

   a. Compute block_number = address / block_size

   b. Compute set_index = block_number % num_sets

   c. Compute tag = block_number / num_sets

   d. Check if tag exists in cache[set_index]:

   - If tag is present:

     Print "HIT"

     Continue to next address

- Else (tag not present):

     Print "MISS"

   e. On MISS:

i. If an empty block (None) exists in the set:

   - Find first empty block index
   - Insert tag into that block

- Print "Stored in empty block"

ii. Else (all blocks full):

- Replace block at position next_replace[set_index]
- Insert new tag into that block
- Print "Replaced block X"
- Update FIFO pointer:

  next_replace[set_index] =

  (next_replace[set_index] + 1) mod associativity

7. Repeat for all addresses.

8. End