REPORT : LAB 6
NAME: P.YASASWINI
ROLL NO: CS22BTECH11046

1. I have create a struct set, in where we will have tag,valid,time

2. int main()

   a. Read the file(cache.config) and stored in respective variables

   b. Now I have calculated 'numsets', block offset 'offset' and 'set_offset'.

   c. Created a 2d array name cache of 'numsets * associativity' where I will store each addressses tag,valid and time.

   d. Dynamically allocated and Initialized them to 0 .

   e. After we will open a file and read line by line, store it a mode and line variables .

   f. calculated set_index and tag (they are seperate variables) and a global variable called hit_check is initialized to 0.

   g. Now we will send all addresses in "cachesearch" function there we will update tag,valid,time and hit_check.

   h. If hit_check is still '0' then it is miss if it is 1 then it is cache hit(* hit_check is a global variable).

   f. Output will be in main function by knowing if address is hit/miss.

3. cachesearch ()

   a. Declared a variable called "check" to check if the respective set index blocks are filled and address is missed.

   b. if check is true there is no need to impliment replacement policies because either there will be a hit or it will be miss and the space will be there to insert the address to the cache

   c. First 'Read': We will traverse through blocks in respective setindex (for loop will be k times (k = way))

   -> if the tag of that address is equal to the respective setindex_block tag and valid =1 then it is a hit

   we will increase time here ,we will break from loop and check is set to true. Here we will make hit_check =1.

   -> if there is miss and space is there, then we will add that address to cache and set check to true and break from loop(here we will set cache[set_index][i].valid = 1)

   -> if check is still false then we will do replacemnet policies

   -> if it is 'LRU' then we will find block which has minimum time in that respective set index from 'findli' function and replace it.

   ->if it is 'FIFO' then we will find block which has maximum time in that respective set index from 'findfi' function and replace it.

   ->if it is 'RANDOM' then we will get random block from 'Random' function (where we will use rand() to get random number and get remainder by dividing it with associative variable) and replace it.

   -> if it is fully associative we wont use Random function

   d. For 'write' : read and write back is same for write through we wont allocate anything if we miss i.e we wont add that address to cahe.

remaining is same if it is hit we will do what we have did in 'Read'

*************************END OF REPORT**************************