CS416 Project2 – Planning

pthread_yield():
if NOT main:
- change current thread state from RUNNING to READY
- increment thread_counter by 1
- call search_next_thread() (call it T)
- if T is null:
    - Done (do nothing)
- if T is BLOCKED (means search_next_thread() returned a thread blocked by join):
    -
- change T's thread state to RUNNING
- change current thread ID pointer to T
- context switch from current thread to T

pthread_join(thread T2, void **value_ptr):
- if called for first time:
    - run scheduler (until T2 is done)
    - done
- fetch T2 from TCB table structure
- if T2 state is DONE:
    -if value_ptr is not null:
        - get T2's return value from TCB hash table
- else:
    - pthread_yield()

pthread_create():
- create TCB
- make the context
- push thread to data structure (ordered linked list and TCB hash table)

pthread_exit(void *value_ptr):
- change pthread_exit flag to true
- free the stack, free the context (retrieve from TCB hash table)
- store value_ptr (thread's return value) into return-value attribute
- change thread status to DONE

pthread_create_helper():
- calls the thread function
- calls pthread_exit if pthread_exit flag is false

search_next_thread(): (let T be the thread with minimum counter)
- if T is BLOCKED and joined_on is not null:
    - check if T's joined_on thread's state is DONE (retrieve from TCB hash table)

        - if joined_on's state is DONE:
            - fetch joined_on thread's return value if any
            - return T
        - else:
            - look at thread with next min counter
- else:
        - return T