Question 1. (1pt) Which of the following scenarios may cause synchronization errors? Circle the best answer.

(a) Two threads attempting to increment a local variable a.
(b) Two processes attempting to read from the same file data.txt.
(c) Two processes attempting to increment a shared variable c.
(d) Two threads attempting to read from a global variable d.
(e) Two Snapchat apps both attempting to use the same filter.

Question 2. (1pt) When you call `pthread_create()`, what happens? Circle the best answer.

(a) Memory is copied from one thread to another.
(b) The thread argument is cast into a “void *” parameter.
(c) The thread disconnects from its parent thread.
(d) A new thread is created that shares all main thread resources.
(e) The thread detaches and subsequently terminates and returns NULL.
(f) Submitty blows up due to a fork bomb.
Question 3. (7pts) What is the exact terminal output of the code below? Assume that all system and library calls return successfully. Further, assume that child thread IDs are assigned sequentially starting at 777. If multiple outputs are possible, use a diagram to clearly and succinctly show all possibilities.

```c
void * wyd( void * arg )
{
    int * x = (int *)arg;
    unsigned int rc = pthread_self();
    printf( "%u unlucky %d\n", rc, *x );
    return NULL;
}

int main()
{
    pthread_t tid1, tid2;
    int x = 7;
    int rc = pthread_create( &tid1, NULL, wyd, &x );
    x = 13;
    rc = pthread_create( &tid2, NULL, wyd, &x );
    printf( "%d lucky %d\n", rc, x );
    rc = pthread_join( tid2, NULL );
    rc = pthread_join( tid1, NULL );
    return EXIT_SUCCESS;
}
```

Question 4. (1pt) Not counting the main thread, how many child threads are created?