1. **(1pt)** When you call `pthread_join()`, what happens? Circle the best answer.

   (a) The thread argument is cast into a “`void *`” parameter.
   (b) The thread disconnects from its parent thread.
   (c) A new thread is created that shares most of the main thread’s resources.
   (d) An existing thread is acknowledged as being terminated.
   (e) Memory is copied from one thread to another, i.e., they join forces.
   (f) The thread detaches by terminating and returning `NULL`.
   (g) Snapchat

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

   (a) The thread argument is cast into a `void *` parameter.
   (b) The thread disconnects from its parent thread.
   (c) A new thread is created that shares most of the main thread’s resources.
   (d) An existing thread is acknowledged as being terminated.
   (e) Memory is copied from one thread to another, i.e., they join forces.
   (f) The thread detaches by terminating and returning `NULL`.
   (g) Snapchat
3. **(6pts)** What is the exact terminal output of the code below? Assume that all system and library calls return successfully. Also assume that child thread IDs are assigned sequentially starting at 777. Further, assume that the final `sleep()` call gives the child processes enough time to terminate. If multiple outputs are possible, use a diagram to clearly and succinctly show all possibilities.

```c
void * omg( void * arg )
{
    pthread_detach( pthread_self() );
    int q = 8;
    int * x = (int *)arg;
    q -= 2;
    printf( "%u lucky %d %d\n", (unsigned int)pthread_self(), q, *x );
    *x -= q;
    return NULL;
}

int main()
{
    int x = 13;
    pthread_t tid;
    pthread_create( &tid, NULL, omg, &x );
    printf( "first thread is %u\n", (unsigned int)tid );
    pthread_create( &tid, NULL, omg, &x );
    printf( "second thread is %u\n", (unsigned int)tid );
    sleep( 10 );
    return EXIT_SUCCESS;
}
```

4. **(2pts)** How would the terminal output change if `sleep()` is removed from the above code?