QUIZ-04

- K22 – V. Atlidakis

Name and student id ____

```
#include <stdio.h>
#include <pthread.h>
#include <time.h>
int *p = NULL;
void *foo(void *arg) {
        printf("%d\n", *p);
}
void main(int argc, char *argv[]) {
        int x = 3;
        pthread_t p1;
        pthread_create(&p1, NULL, foo, (void *)&x);
        p = &x;
}
```

1) What are all the possible things that **could** "happen" as a result of executing the above program? Enumerate.

- Print 3, exit without printing anything, or crash.

2) The **usleep()** function suspends execution of the calling thread for (at least) usec microseconds. Use it above as a synchronization primitive to help **increase the likelihood** that the above program outputs "3".

- Sleep before printf in the secondary thread to give enough time to the main thread to assign a valid, non-null address to the pointer, and sleep again before the end of main. Note that the second sleep needs to be meaningfully larger than the first to give a better chance of ordering the operations as intended.

3) Is the above "synchronization primitive" robust? Explain your answer.

- No. A myriad of different possible interleavings, dependent on various external asynchronous events, may lead to unintended orders of execution.