CSSE 132 – Introduction to Systems Programming
Rose-Hulman Institute of Technology

Worksheet 25 Parallel Programming with `pthread`

Name (Print):_____ RHIT Username:_____

1. Read the code below and answer the following questions.

```
1  #include <stdio.h>
2  #include <pthread.h>
3
4  void* myThreadFun(void* argp) {
5    int* ptr = (int*) argp;
6    int tid = *ptr;
7    printf("I'm thread %d\n", tid);
8  }
9
10 void main() {
11   pthread_t t0, t1;
12   int a = 1, b = 2;
13   pthread_create(&t0, NULL, myThreadFun, &a);
14   pthread_create(&t1, NULL, myThreadFun, &b);
15   pthread_join(t0, NULL);
16   pthread_join(t1, NULL);
17 }
```

   (a) What happens when calling `pthread_create(&t0, NULL, myThreadFun, &a);`?

   It starts the [ ] `t0` *by invoking* `myThreadFun` *with* [ ] `&a`.

   (b) What are the **data types** of the input arguments of `pthread_create`?

   The 1st argument: [ ] The 4th argument: [ ]

   (c) Look at Lines 5-6 to understand how the variables `a` and `b` in `main` are passed to different threads. If a thread function requires more than one variable (say a `float` and `int`), how can you implement it so that different threads can take different inputs?

   (d) What happens when we run `pthread_join(t0, NULL);`?

2. **Parallel Calculations Using Pthreads**

Complete the four blanks so that three threads calculate the squares of 3, 5, and 7. Assume all thread creation and join calls succeed.

```c
#include <stdio.h>
#include <pthread.h>

void *square(void *arg) {
    int num = __________;

    printf("%d squared = %d\n", num, num * num);

    return NULL;
}

int main(void) {
    pthread_t threads[3];
    int numbers[3] = {3, 5, 7};

    // Create three threads
    for (int i = 0; i < 3; i++) {
        pthread_create(__________, NULL,
                       square, __________);
    }

    // Wait for all threads to finish
    for (int i = 0; i < 3; i++) {
        pthread_join(__________, NULL);
    }

    printf("All done!\n");
    return 0;
}
```

Question 2 continued: Explain your reasoning.

1. Why do we pass `&numbers[i]` instead of `numbers[i]`?

2. Why are all threads created before calling `pthread_join()`?

3. Will the squared results always appear in the same order? Explain.

4. What might happen if we remove the `pthread_join()` calls?

3. **Reflection:** What is one thing from this topic that you found interesting, OR what is one question you are now curious to learn more about?