Names: _____

On my honor, I/we have not given, nor received,
nor witnessed any unauthorized assistance on this work.

Signature: _____

Signature: _____

Signature: _____

Signature: _____

Signature: _____

Collaboration Stmt - Reference any resources (beyond those listed on the Sprint Resources page) which you used in completing this assignment:

Question:	1	2	3	4	5	Total
Points:	22	10	10	8	10	60
Score:						

TOTAL GRADE: _____/ 100%

1. (22 points) The following timeline shows an example of a process being created, executing, issuing a system call, and exiting. Each step in the timeline is done by one of three entities: the OS, the HW, or the USER (ie, the user program itself). For each step, state who is responsible.

create entry for process list	_____
allocate memory for program	_____
load program into memory	_____
setup user stack with `argv`	_____
fill kernel stack with registers and PC	_____
execute `return-from-trap` instruction	_____
restore registers from kernel stack	_____
switch to user mode	_____
set PC to `main()`	_____
start running in `main()`	_____
call a system call	_____
execute a trap instruction	_____
save registers to kernel stack	_____
switch to kernel mode	_____
set PC to OS trap handler	_____
handle trap	_____
do the work of the syscall	_____
execute `return-from-trap` instruction	_____
restore registers from kernel stack	_____
switch to user mode	_____
set PC to instruction after earlier trap	_____
call `exit()` system call	_____

2. (10 points) Here are two programs named `decrement.c` (left) and `reset.c` (right).

```
int value = 10;
int main(void) {
    while (1) {
        printf("%d", value);
        value--;
    }
    return 0;
}
```

```
int value;
int main(void) {
    value = 0;
    return 0;
}
```

While `decrement.c` is running, `reset.c`, is run once as a separate process. You see a series of numbers as output to the terminal.

State whether each of the following outputs are possible or not possible given the scenario above. If an output is not possible give a brief explanation why not.

(a) `1098765 ...`

(b) `1098710987 ...`

(c) `109876510987 ...`

(d) `109876543210 ...`

(e) `987654 ...`

3. (10 points) Assume we have only one CPU and we are currently using a FIFO scheduler. Three jobs arrive (in the order listed), and each job has a required runtime, which means the job needs that many time units on the CPU to complete.

```
Job A arrives at time=0, required runtime=X time units
Job B arrives at time=5, required runtime=Y time units
Job C arrives at time=10, required runtime=Z time units
```

Assuming an **average turnaround time** between 10 and 20 time units (inclusive), which of the following run times for A, B, and C are possible? Briefly justify your answer (this can be a mathematical justification).

(a) X=10, Y=10, Z=10

(b) X=20, Y=20, Z=20

(c) X=5, Y=10, Z=15

(d) X=20, Y=30, Z=40

4. (8 points) Assume the following schedule for a set of three jobs, A, B, and C:

```
A runs first (for 10 time units) but is not yet done
B runs next (for 10 time units) but is not yet done
C runs next (for 10 time units) and runs to completion
A runs to completion (for 10 time units)
B runs to completion (for 5 time units)
```

For each scheduling scheme, state whether or not the above sequence could occur. Briefly justify/explain your answer.

(a) FIFO

(b) Round Robin

(c) STCF (Shortest Time to Completion First)

(d) Multi-level Feedback Queue

5. (10 points) The Multi-level Feedback Queue (MLFQ) is a fancy scheduler that does lots of things. Label each of the following statements about the MLFQ as True or False and briefly justify your answer.

   (a) MLFQ learns things about running jobs

   (b) MLFQ starves long running jobs

   (c) MLFQ uses different length time slices for jobs

   (d) MLFQ uses round robin

   (e) MLFQ forgets what it has learned about running jobs sometimes