

1. Sorting Swap bug.

- Original Code had a mistake inside the SortByArrival function.

```
p[i] = [j];  
p[j] = temp;
```

- Compiler said “expected identifier before ‘[‘ token.”
- Replaced it with the proper swap logic

```
Process temp = p[i];
```

```
p[i] = p[j]  
p[j] = temp
```

2. Results table printed inside the loop.

- It put the results table inside of the FCFS loop, the output made a full table after each process instead of just printing one table, so I moved the whole table section outside of the loop to make it just one table with the completed information.

```
for (int i = 0; i < n; ++i) {  
    cout << setw(5) << p[i].pid  
    << setw(10) << p[i].arrival  
    << setw(8) << p[i].burst  
    << setw(8) << p[i].start  
    << setw(8) << p[i].finish  
    << setw(10) << p[i].waiting  
    << setw(12) << p[i].turnaround  
    << '\n';  
}
```

3. Missing the finish/wait/turnaround calculation

- I forgot to set the finish, waiting, and turnaround values after completing the code. So when the tables were made, it either had inaccurate values or zeroes. So I used the pseudocode and algorithms we made from the slides, and just did the standard calculations for the FCFS algorithms and put it in the code

```
p[i].finish = current_time;  
P[i].waiting = p[i].start - p[i].arrival;
```

```
P[i].turnaround = p[i].finish - p[i].arrival.
```

4. Missing Average accumulations

- Average waiting time and turnarounds were printed as 0.0 once the tables came around. I added an accumulation inside the loop so it would constantly update and give an accurate average waiting time.

```
total_wait += p[i].waiting  
total_turn += p[i].turnaround;
```

5. CPU Idle section bugs.

- The first time the CPU idle test was going, it skipped the idle cycles in the first versions. It didn't show a CPU idle time even though the arrival times were far apart. So I added a while loop that accumulated the current time to display it correctly.

```
while (current_time < p[i].arrival) {  
    cout << "Time " << current_time << ": CPU idle\n";  
    current_time++;  
}
```