SOLUTION:5B

```
#include <stdio.h>

int main() {
    int n = 9; // We are sorting 9 processes
    int i, j, temp, time = 0;
    int arrival[9], burst[9], waiting[9], turnaround[9], completion[9], process[9];
    float totalWaiting = 0, totalTurnaround = 0;

    // Input arrival time and burst time for 9 processes
    printf("Enter the arrival time and burst time for 9 processes:\n");
    for (i = 0; i < n; i++) {
        printf("Process %d Arrival Time: ", i + 1);
        scanf("%d", &arrival[i]);
        printf("Process %d Burst Time: ", i + 1);
        scanf("%d", &burst[i]);
        process[i] = i + 1; // Process ID
    }

    // Sort processes based on burst time (SJF), using a simple bubble sort
    for (i = 0; i < n; i++) {
        for (j = i + 1; j < n; j++) {
            if (burst[i] > burst[j]) {
                // Swap burst time
                temp = burst[i];
                burst[i] = burst[j];
                burst[j] = temp;

                // Swap arrival time accordingly
                temp = arrival[i];
                arrival[i] = arrival[j];
                arrival[j] = temp;

                // Swap process IDs accordingly
                temp = process[i];
                process[i] = process[j];
                process[j] = temp;
            }
        }
    }

    // Calculate completion time, turnaround time, and waiting time
    for (i = 0; i < n; i++) {
        if (time < arrival[i]) {
            time = arrival[i]; // If the CPU is idle, wait for the next process
        }
        time += burst[i]; // Increase the time by burst time of the selected process
        completion[i] = time; // Completion time is when the process finishes
        turnaround[i] = completion[i] - arrival[i]; // Turnaround time = Completion time - Arrival time
        waiting[i] = turnaround[i] - burst[i]; // Waiting time = Turnaround time - Burst time
        totalWaiting += waiting[i]; // Add to total waiting time
        totalTurnaround += turnaround[i]; // Add to total turnaround time
    }

    // Display the sorted processes and their corresponding times
    printf("\nProcess\tArrival\tBurst\tCompletion\tWaiting\tTurnaround\n");
    for (i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\t\t%d\t%d\n", process[i], arrival[i], burst[i], completion[i], waiting[i],
turnaround[i]);
    }
    // Display average waiting and turnaround times
    printf("\nAverage Waiting Time: %.2f\n", totalWaiting / n);
    printf("Average Turnaround Time: %.2f\n", totalTurnaround / n);

    return 0;
}
```