6a.

CODE-

cse16@localhost:~

```
#include <stdio.h>

int main() {
    int n, i;
    int burst_time[10], waiting_time[10], turnaround_time[10];
    int total_waiting_time = 0, total_turnaround_time = 0;
    float avg_waiting_time, avg_turnaround_time;

    // Step 1: Get the number of processes from the user
    printf("Enter the number of processes: ");
    scanf("%d", &n);

    // Step 2: Read the burst times for each process
    printf("Enter the burst time of the processes:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &burst_time[i]);
    }

    // Step 3: Calculate the waiting time and turnaround time for each process
    waiting_time[0] = 0; // Waiting time for the first process is 0

    // Calculate waiting time for all processes except the first one
    for (i = 1; i < n; i++) {
        waiting_time[i] = burst_time[i - 1] + waiting_time[i - 1];
    }

    // Calculate turnaround time and total waiting time, total turnaround time
    for (i = 0; i < n; i++) {
        turnaround_time[i] = burst_time[i] + waiting_time[i];
        total_waiting_time += waiting_time[i];
        total_turnaround_time += turnaround_time[i];
    }

    // Step 4: Display the process name, burst time, waiting time, and turnaround time
    printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
    for (i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\n", i, burst_time[i], waiting_time[i], turnaround_time[i]);
    }

    // Step 5: Calculate and display the average waiting time and average turnaround time
    avg_waiting_time = (float)total_waiting_time / n;
    avg_turnaround_time = (float)total_turnaround_time / n;

    printf("\nAverage waiting time is: %.2f", avg_waiting_time);
    printf("\nAverage Turnaround Time is: %.2f", avg_turnaround_time);

    return 0;
}
```

OUTPUT-

```
Enter the number of processes: 3
Enter the burst time of the processes:
24 3 3

Process Burst Time      Waiting Time    Turnaround Time
0       24              0               24
1       3               24              27
2       3               27              30

Average waiting time is: 17.00
Average Turnaround Time is: 27.00[cse16@localhost ~]$ ^C
```

6b.

CODE-

cse16@localhost:~

```c
#include <stdio.h>
struct Process {
    int process_id;
    int burst_time;
    int waiting_time;
    int turn_around_time;
};

void calculateTimes(struct Process p[], int n) {
    int total_waiting_time = 0, total_turn_around_time = 0;
    for (int i = 0; i < n; i++) {
        p[i].waiting_time = (i == 0) ? 0 : p[i - 1].waiting_time + p[i - 1].burst_time;
        p[i].turn_around_time = p[i].waiting_time + p[i].burst_time;

        total_waiting_time += p[i].waiting_time;
        total_turn_around_time += p[i].turn_around_time;
    }
    float avg_waiting_time = (float)total_waiting_time / n;
    float avg_turn_around_time = (float)total_turn_around_time / n;
    printf("\nProcess\tBurst Time\tWaiting Time\tTurn Around Time\n");
    for (int i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\n", p[i].process_id, p[i].burst_time, p[i].waiting_time, p[i].turn_around_time);
    }

    printf("\nAverage waiting time: %.2f", avg_waiting_time);
    printf("\nAverage turn around time: %.2f\n", avg_turn_around_time);
}
int main() {
    int n;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    struct Process p[n];
    printf("Enter the burst time of the processes:\n");
    for (int i = 0; i < n; i++) {
        p[i].process_id = i + 1;
        scanf("%d", &p[i].burst_time);
    }

    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (p[i].burst_time > p[j].burst_time) {
                struct Process temp = p[i];
                p[i] = p[j];
                p[j] = temp;
            }
        }
    }
    calculateTimes(p, n);

    return 0;
}
```

OUTPUT-

```
Enter the number of processes: 4
Enter the burst time of the processes:
8 4 9 5

Process Burst Time      Waiting Time    Turn Around Time
2       4               0               4
4       5               4               9
1       8               9               17
3       9               17              26

Average waiting time: 7.50
Average turn around time: 14.00
```

6c.

CODE-

cse16@localhost:~

```c
#include <stdio.h>
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
int main() {
    int n;
    printf("Enter Total Number of Processes: ");
    scanf("%d", &n);
    int burstTime[n], priority[n], processId[n];
    for (int i = 0; i < n; i++) {
        printf("Enter Burst Time and Priority for P[%d]:\n", i + 1);
        printf("Burst Time: ");
        scanf("%d", &burstTime[i]);
        printf("Priority: ");
        scanf("%d", &priority[i]);
        processId[i] = i + 1;
    }
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (priority[j] > priority[j + 1]) {
                swap(&priority[j], &priority[j + 1]);
                swap(&burstTime[j], &burstTime[j + 1]);
                swap(&processId[j], &processId[j + 1]);
            }
        }
    }
    int waitingTime[n], turnaroundTime[n];
    waitingTime[0] = 0;
    turnaroundTime[0] = burstTime[0];
    for (int i = 1; i < n; i++) {
        waitingTime[i] = waitingTime[i - 1] + burstTime[i - 1];
        turnaroundTime[i] = waitingTime[i] + burstTime[i];
    }
    printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
    for (int i = 0; i < n; i++) {
        printf("P[%d]\t%d\t\t%d\t\t%d\n", processId[i], burstTime[i], waitingTime[i], turnaroundTime[i]);
    }
    float avgWaitingTime = 0, avgTurnaroundTime = 0;
    for (int i = 0; i < n; i++) {
        avgWaitingTime += waitingTime[i];
        avgTurnaroundTime += turnaroundTime[i];
    }
    avgWaitingTime /= n;
    avgTurnaroundTime /= n;
    printf("\nAverage Waiting Time = %.2f\n", avgWaitingTime);
    printf("Average Turnaround Time = %.2f\n", avgTurnaroundTime);

    return 0;
}
```

OUTPUT-

```
Enter Total Number of Processes: 4
Enter Burst Time and Priority for P[1]:
Burst Time: 6
Priority: 3
Enter Burst Time and Priority for P[2]:
Burst Time: 2
Priority: 2
Enter Burst Time and Priority for P[3]:
Burst Time: 14
Priority: 1
Enter Burst Time and Priority for P[4]:
Burst Time: 6
Priority: 4

Process Burst Time      Waiting Time    Turnaround Time
P[3]    14              0               14
P[2]    2               14              16
P[1]    6               16              22
P[4]    6               22              28

Average Waiting Time = 13.00
Average Turnaround Time = 20.00
[csel6@localhost ~]$
```

6d.

CODE-

cse16@localhost:~

```
int main() {
    int n, quantum;
    printf("Enter Total Number of Processes: ");
    scanf("%d", &n);
    int pid[n], arrival_time[n], burst_time[n], remaining_burst_time[n], waiting_time[n], turnaround_time[n];
    for (int i = 0; i < n; i++) {
        pid[i] = i + 1;
        printf("Enter Details of Process [%d]:\n", i + 1);
        printf("Arrival Time: ");
        scanf("%d", &arrival_time[i]);
        printf("Burst Time: ");
        scanf("%d", &burst_time[i]);
        remaining_burst_time[i] = burst_time[i];
        waiting_time[i] = 0;
        turnaround_time[i] = 0;
    }
    printf("Enter Time Quantum: ");
    scanf("%d", &quantum);
    int time = 0;
    int remaining_processes = n;
    while (remaining_processes != 0) {
        for (int i = 0; i < n; i++) {
            if (remaining_burst_time[i] > 0) {
                if (remaining_burst_time[i] <= quantum) {
                    time += remaining_burst_time[i];
                    waiting_time[i] = time - burst_time[i] - arrival_time[i];
                    remaining_burst_time[i] = 0;
                    turnaround_time[i] = time - arrival_time[i];
                    remaining_processes--;
                } else {
                    time += quantum;
                    remaining_burst_time[i] -= quantum;
                }
            }
        }
    }
    float total_waiting_time = 0, total_turnaround_time = 0;
    printf("\nProcess ID\tBurst Time\tTurnaround Time\tWaiting Time\n");
    for (int i = 0; i < n; i++) {
        printf("Process [%d]\t%d\t\t%d\t\t%d\n", pid[i], burst_time[i], turnaround_time[i], waiting_time[i]);
        total_waiting_time += waiting_time[i];
        total_turnaround_time += turnaround_time[i];
    }
    printf("\nAverage Waiting Time: %.6f\n", total_waiting_time / n);
    printf("Average Turnaround Time: %.6f\n", total_turnaround_time / n);

    return 0;
}
```

OUTPUT-

```
Enter Total Number of Processes: 4
Enter Details of Process [1]:
Arrival Time: 0
Burst Time: 4
Enter Details of Process [2]:
Arrival Time: 1
Burst Time: 7
Enter Details of Process [3]:
Arrival Time: 2
Burst Time: 5
Enter Details of Process [4]:
Arrival Time: 3
Burst Time: 6
Enter Time Quantum: 1

Process ID      Burst Time      Turnaround Time Waiting Time
Process [1]     4               13              9
Process [2]     7               21              14
Process [3]     5               16              11
Process [4]     6               18              12

Average Waiting Time: 11.500000
Average Turnaround Time: 17.000000
```