```c
#include <stdio.h>

#define MAX 10

typedef struct {
  int id;
  int arrival;
  int burst;
  int priority;
  int waiting;
  int turnaround;
} Process;

void swap(Process *a, Process *b) {
  Process temp = *a;
  *a = *b;
  *b = temp;
}

void sortByArrival(Process proc[], int n) {
  for (int i = 0; i < n - 1; i++) {
    for (int j = i + 1; j < n; j++) {
      if (proc[i].arrival > proc[j].arrival) {
        swap(&proc[i], &proc[j]);
      }
    }
  }
}

void sortByBurst(Process proc[], int n) {
  for (int i = 0; i < n - 1; i++) {
    for (int j = i + 1; j < n; j++) {
      if (proc[i].burst > proc[j].burst) {
        swap(&proc[i], &proc[j]);
      }
    }
  }
}

void sortByPriority(Process proc[], int n) {
  for (int i = 0; i < n - 1; i++) {
    for (int j = i + 1; j < n; j++) {
      if (proc[i].priority > proc[j].priority) {
        swap(&proc[i], &proc[j]);
      }
    }
  }
}

void FCFS(Process proc[], int n) {
  int total_wt = 0, total_tat = 0;
  proc[0].waiting = 0;
  proc[0].turnaround = proc[0].burst;

  for (int i = 1; i < n; i++) {
    proc[i].waiting = proc[i - 1].waiting + proc[i - 1].burst - proc[i].arrival;
    if (proc[i].waiting < 0)
      proc[i].waiting = 0;
    proc[i].turnaround = proc[i].waiting + proc[i].burst;
    total_wt += proc[i].waiting;
    total_tat += proc[i].turnaround;
  }
  printf("\nFCFS Scheduling:\n");
  for (int i = 0; i < n; i++) {
    printf("Process %d: Waiting Time = %d, Turnaround Time = %d\n", proc[i].id,
           proc[i].waiting, proc[i].turnaround);
  }
  printf("Average Waiting Time = %.2f\n", (float)total_wt / n);
  printf("Average Turnaround Time = %.2f\n", (float)total_tat / n);
}

void SJF(Process proc[], int n) {
  int total_wt = 0, total_tat = 0;
  int time = 0;
  proc[0].waiting = 0;
  proc[0].turnaround = proc[0].burst;

  for (int i = 1; i < n; i++) {
    proc[i].waiting = time - proc[i].arrival;
    if (proc[i].waiting < 0)
      proc[i].waiting = 0;
    proc[i].turnaround = proc[i].waiting + proc[i].burst;
    total_wt += proc[i].waiting;
    total_tat += proc[i].turnaround;
    time += proc[i].burst;
  }

  printf("\nSJF Scheduling:\n");
  for (int i = 0; i < n; i++) {
    printf("Process %d: Waiting Time = %d, Turnaround Time = %d\n", proc[i].id,
           proc[i].waiting, proc[i].turnaround);
  }
  printf("Average Waiting Time = %.2f\n", (float)total_wt / n);
  printf("Average Turnaround Time = %.2f\n", (float)total_tat / n);
}

void PriorityScheduling(Process proc[], int n) {
  int total_wt = 0, total_tat = 0;
  int time = 0;
  proc[0].waiting = 0;
  proc[0].turnaround = proc[0].burst;

  for (int i = 1; i < n; i++) {
    proc[i].waiting = time - proc[i].arrival;
    if (proc[i].waiting < 0)
      proc[i].waiting = 0;
    proc[i].turnaround = proc[i].waiting + proc[i].burst;
    total_wt += proc[i].waiting;
    total_tat += proc[i].turnaround;
    time += proc[i].burst;
  }

  printf("\nPriority Scheduling:\n");
  for (int i = 0; i < n; i++) {
    printf("Process %d: Waiting Time = %d, Turnaround Time = %d\n", proc[i].id,
           proc[i].waiting, proc[i].turnaround);
  }
  printf("Average Waiting Time = %.2f\n", (float)total_wt / n);
  printf("Average Turnaround Time = %.2f\n", (float)total_tat / n);
}

int main() {
  Process proc[MAX];
  int n;

  printf("Enter number of processes: ");
  scanf("%d", &n);

  for (int i = 0; i < n; i++) {
    printf("Enter arrival time, burst time, and priority for process %d: ",
           i + 1);
    proc[i].id = i + 1;
    scanf("%d %d %d", &proc[i].arrival, &proc[i].burst, &proc[i].priority);
  }

  sortByArrival(proc, n);
  FCFS(proc, n);

  sortByArrival(proc, n);
  sortByBurst(proc, n);
  SJF(proc, n);

  sortByArrival(proc, n);
  sortByPriority(proc, n);
  PriorityScheduling(proc, n);

  return 0;
}
```