```
#include <stdio.h>

// Structure to represent a process
struct Process {
    int id;           // Process ID
    int arrivalTime;  // Arrival time
    int burstTime;    // Burst time
    int priority;     // Priority
    int remainingTime;  // Remaining burst time
};

// Function to select the process with the highest priority
int selectProcess(struct Process processes[], int n, int currentTime) {
    int highestPriority = -1;
    int highestPriorityIndex = -1;

    // Find the process with the highest priority among arrived processes
    for (int i = 0; i < n; i++) {
        if (processes[i].arrivalTime <= currentTime && processes[i].remainingTime > 0) {
            if (highestPriority == -1 || processes[i].priority < highestPriority) {
                highestPriority = processes[i].priority;
                highestPriorityIndex = i;
            }
        }
    }

    return highestPriorityIndex;
}

int main() {
    int n; // Number of processes
```

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

struct Process processes[n]; // Array to store processes

// Input the process details
for (int i = 0; i < n; i++) {
    printf("Enter arrival time for process %d: ", i + 1);
    scanf("%d", &processes[i].arrivalTime);
    printf("Enter burst time for process %d: ", i + 1);
    scanf("%d", &processes[i].burstTime);
    printf("Enter priority for process %d: ", i + 1);
    scanf("%d", &processes[i].priority);
    processes[i].id = i + 1;
    processes[i].remainingTime = processes[i].burstTime;
}

int currentTime = 0;
printf("\nGantt Chart:\n");
printf("----------\n");

// Perform scheduling
while (1) {
    int selectedProcessIndex = selectProcess(processes, n, currentTime);

    if (selectedProcessIndex == -1)
        break; // No processes remaining

    struct Process *selectedProcess = &processes[selectedProcessIndex];
    printf("| P%d ", selectedProcess->id);
```

```
        // Update remaining time and current time
        selectedProcess->remainingTime--;
        currentTime++;

        // Check if process is complete
        if (selectedProcess->remainingTime == 0) {
            printf("|");
        }
    }

    printf("\n\n");

    return 0;
}
```