

**5. Construct a scheduling program with C that selects the waiting process with the highest priority to execute next.**

```
#include <stdio.h>
int main() {
    int n;
    printf("Enter number of processes: ");
    scanf("%d", &n);

    int bt[n], pr[n], id[n];
    for(int i = 0; i < n; i++) {
        id[i] = i + 1; // process IDs
        printf("Enter burst time of P%d: ", i+1);
        scanf("%d", &bt[i]);
        printf("Enter priority of P%d (lower number = higher priority): ", i+1);
        scanf("%d", &pr[i]);
    }

    // Sort processes based on priority (non-preemptive)
    for(int i = 0; i < n-1; i++) {
        int min_idx = i;
        for(int j = i+1; j < n; j++) {
            if(pr[j] < pr[min_idx]) { // higher priority
                min_idx = j;
            }
        }
        // Swap priority
        int temp = pr[i]; pr[i] = pr[min_idx]; pr[min_idx] = temp;
        // Swap burst time
        temp = bt[i]; bt[i] = bt[min_idx]; bt[min_idx] = temp;
        // Swap process ID
        temp = id[i]; id[i] = id[min_idx]; id[min_idx] = temp;
    }

    // Calculate waiting time and turnaround time
    int wt[n], tat[n];
    wt[0] = 0;
    tat[0] = bt[0];
    for(int i = 1; i < n; i++) {
        wt[i] = wt[i-1] + bt[i-1];
        tat[i] = wt[i] + bt[i];
    }

    printf("\nProcess\tBurst\tPriority\tWaiting\tTurnaround\n");
    for(int i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\t%d\n", id[i], bt[i], pr[i], wt[i], tat[i]);
    }
}
```

```

    }

    // Calculate average waiting and turnaround time
    double awt=0, atat=0;
    for(int i = 0; i < n; i++) {
        awt += wt[i];
        atat += tat[i];
    }
    printf("\nAverage Waiting Time = %.2f", awt/n);
    printf("\nAverage Turnaround Time = %.2f\n", atat/n);

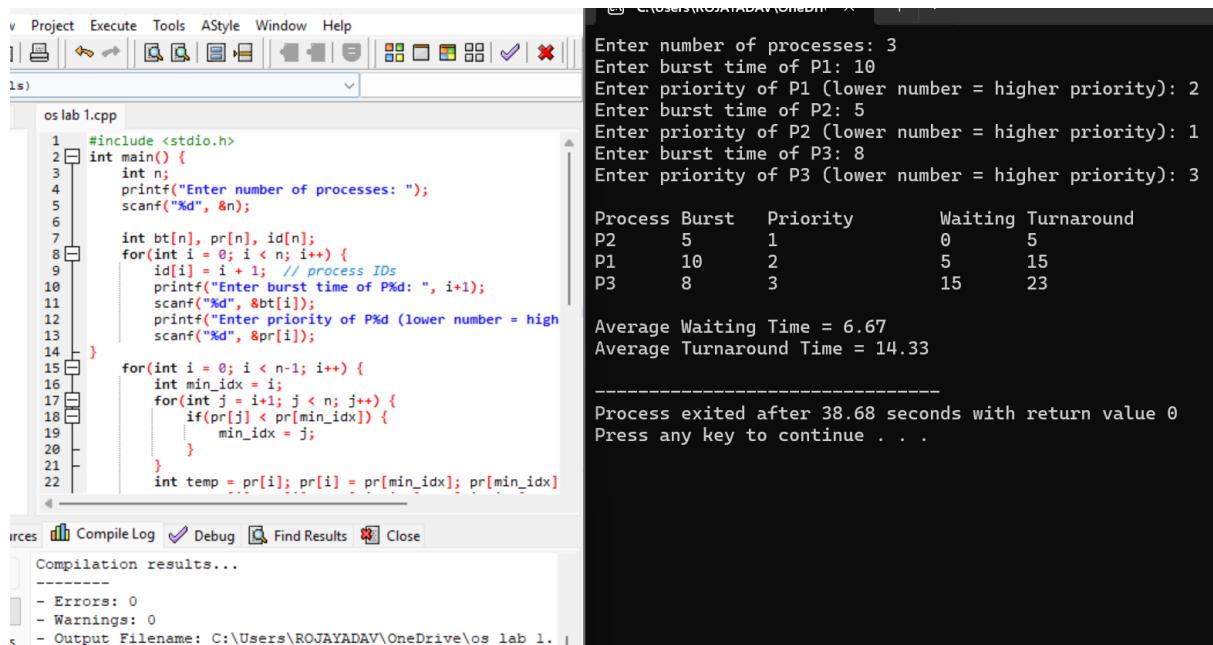
    return 0;
}

```

### INPUT:

Enter number of processes: 3  
 Enter burst time of P1: 10  
 Enter priority of P1 (lower number = higher priority): 2  
 Enter burst time of P2: 5  
 Enter priority of P2 (lower number = higher priority): 1  
 Enter burst time of P3: 8  
 Enter priority of P3 (lower number = higher priority): 3

### OUTPUT:



The screenshot shows a C++ development environment with the following details:

- Source Code (os lab 1.cpp):**

```

1 #include <stdio.h>
2 int main() {
3     int n;
4     printf("Enter number of processes: ");
5     scanf("%d", &n);
6
7     int bt[n], pr[n], id[n];
8     for(int i = 0; i < n; i++) {
9         id[i] = i + 1; // process IDs
10        printf("Enter burst time of P%d: ", i+1);
11        scanf("%d", &bt[i]);
12        printf("Enter priority of P%d (lower number = high");
13        scanf("%d", &pr[i]);
14    }
15    for(int i = 0; i < n-1; i++) {
16        int min_idx = i;
17        for(int j = i+1; j < n; j++) {
18            if(pr[j] < pr[min_idx]) {
19                min_idx = j;
20            }
21        }
22        int temp = pr[i]; pr[i] = pr[min_idx]; pr[min_idx] = temp;
    }
}

```
- Output Window:**

```

Enter number of processes: 3
Enter burst time of P1: 10
Enter priority of P1 (lower number = higher priority): 2
Enter burst time of P2: 5
Enter priority of P2 (lower number = higher priority): 1
Enter burst time of P3: 8
Enter priority of P3 (lower number = higher priority): 3

Process   Burst      Priority      Waiting   Turnaround
P2        5          1             0          5
P1        10         2             5          15
P3        8          3             15         23

Average Waiting Time = 6.67
Average Turnaround Time = 14.33

-----
Process exited after 38.68 seconds with return value 0
Press any key to continue . .

```
- Compiler Output (Compile Log):**
  - Compilation results...
  - Errors: 0
  - Warnings: 0
  - Output Filename: C:\Users\ROJAYADAV\OneDrive\os lab 1.