EXP 4: Construct a scheduling program with C that selects the waiting process with the smallest execution time to execute next.

```
#include <stdio.h>

int main() {
    int n, i, j;
    int bt[20], p[20], wt[20], tat[20];
    int temp;
    float total_wt = 0, total_tat = 0;

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

    // Input burst times and assign process numbers
    for (i = 0; i < n; i++) {
        printf("Enter burst time for process %d: ", i + 1);
        scanf("%d", &bt[i]);
        p[i] = i + 1; // Process number
    }

    // Sort processes by burst time (SJF)
    for (i = 0; i < n - 1; i++) {
        for (j = i + 1; j < n; j++) {
            if (bt[j] < bt[i]) {
                // Swap burst times
                temp = bt[i];
                bt[i] = bt[j];
                bt[j] = temp;

                // Swap process numbers
```

```
            temp = p[i];
            p[i] = p[j];
            p[j] = temp;
        }
    }
}

// Calculate waiting time
wt[0] = 0;
for (i = 1; i < n; i++) {
    wt[i] = wt[i - 1] + bt[i - 1];
    total_wt += wt[i];
}

// Calculate turnaround time
for (i = 0; i < n; i++) {
    tat[i] = wt[i] + bt[i];
    total_tat += tat[i];
}

// Print results
printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
for (i = 0; i < n; i++) {
    printf("P%d\t%d\t\t%d\t\t%d\n", p[i], bt[i], wt[i], tat[i]);
}

printf("\nAverage Waiting Time = %.2f", total_wt / n);
printf("\nAverage Turnaround Time = %.2f\n", total_tat / n);

return 0;
}
```

Sample Input

Enter number of processes: 4

Enter burst time for process 1: 2

Enter burst time for process 2: 4

Enter burst time for process 3: 5

Enter burst time for process 4: 6

Sample Output

```
Process Burst Time      Waiting Time    Turnaround Time
P1      2               0               2
P2      4               2               6
P3      5               6               11
P4      6               11              17

Average Waiting Time = 4.75
Average Turnaround Time = 9.00
```