NAME : HUZAIFA BIN SALMAN

ROLL NO : DT-22034

Round Robin Scheduling

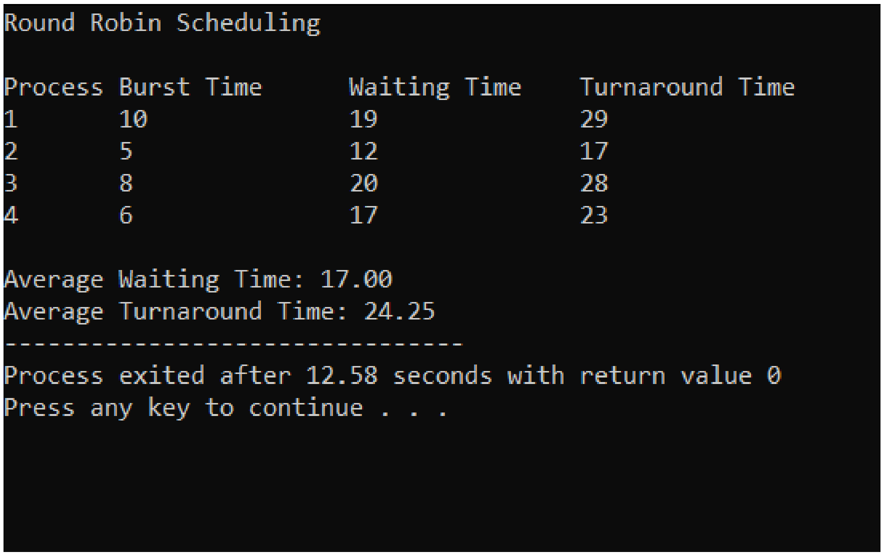
CODE :

```
#include <stdio.h>
```

```
void findWaitingTime(int processes[], int n, int bt[], int wt[], int quantum) { int rem_bt[n]; for (int i = 0; i < n; i++) rem_bt[i] = bt[i];
```

```
int t = 0;
while (1) {
    int done = 1;
    for (int i = 0; i < n; i++) {
        if (rem_bt[i] > 0) {
            done = 0;
            if (rem_bt[i] > quantum) {
                t += quantum;
                rem_bt[i] -= quantum;
            } else {
                t += rem_bt[i];
                wt[i] = t - bt[i];
                rem_bt[i] = 0;
            }
        }
    }
    if (done) break;
}


}
```

```
void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) { for (int i = 0; i < n; i++) { tat[i] = bt[i] + wt[i]; } }
void findAvgTime(int processes[], int n, int bt[], int quantum) { int wt[n], tat[n];
findWaitingTime(processes, n, bt, wt, quantum); findTurnAroundTime(processes, n, bt, wt,
tat);

int total_wt = 0, total_tat = 0;
printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
for (int i = 0; i < n; i++) {
    total_wt += wt[i];
    total_tat += tat[i];
    printf("%d\t%d\t\t%d\t\t%d\n", processes[i], bt[i], wt[i],
tat[i]);
}

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


}

int main() { int processes[] = {1, 2, 3, 4}; int n = sizeof(processes) / sizeof(processes[0]); int
burst_time[] = {10, 5, 8, 6}; int quantum = 3;

printf("Round Robin Scheduling\n");
findAvgTime(processes, n, burst_time, quantum);

return 0;

}
```



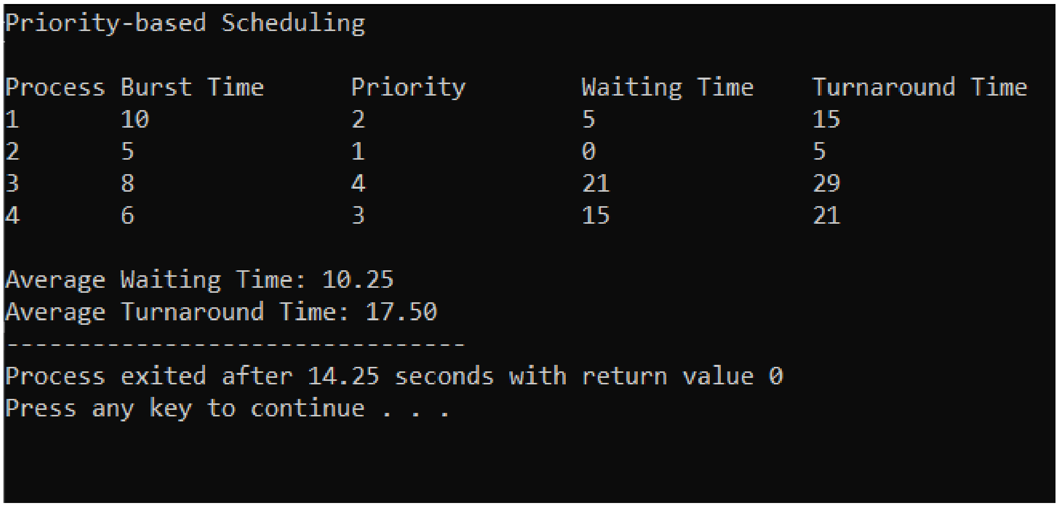
```
Round Robin Scheduling

Process Burst Time      Waiting Time    Turnaround Time
1       10              19              29
2       5               12              17
3       8               20              28
4       6               17              23

Average Waiting Time: 17.00
Average Turnaround Time: 24.25
--------------------------------
Process exited after 12.58 seconds with return value 0
Press any key to continue . . .
```

Priority-based Scheduling

CODE:

```
#include <stdio.h>

void findWaitingTime(int processes[], int n, int bt[], int wt[], int priority[]) { int completed[n],
t = 0; for (int i = 0; i < n; i++) completed[i] = 0;

for (int count = 0; count < n; count++) {
    int max_priority = -1, idx = -1;
    for (int i = 0; i < n; i++) {
        if (!completed[i] && (idx == -1 || priority[i] <
max_priority)) {
            max_priority = priority[i];
            idx = i;
        }
    }
    wt[idx] = t;
    t += bt[idx];
    completed[idx] = 1;
}


}
```

```
void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) { for (int i = 0; i < n;
i++) { tat[i] = bt[i] + wt[i]; } }

void findAvgTime(int processes[], int n, int bt[], int priority[]) { int wt[n], tat[n];
findWaitingTime(processes, n, bt, wt, priority); findTurnAroundTime(processes, n, bt, wt,
tat);

int total_wt = 0, total_tat = 0;
printf("\nProcess\tBurst Time\tPriority\tWaiting Time\tTurnaround
Time\n");
for (int i = 0; i < n; i++) {
    total_wt += wt[i];
    total_tat += tat[i];
    printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", processes[i], bt[i],
priority[i], wt[i], tat[i]);
}

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


}

int main() { int processes[] = {1, 2, 3, 4}; int n = sizeof(processes) / sizeof(processes[0]); int
burst_time[] = {10, 5, 8, 6}; int priority[] = {2, 1, 4, 3};

printf("Priority-based Scheduling\n");
findAvgTime(processes, n, burst_time, priority);

return 0;


}
```



```
Priority-based Scheduling

Process Burst Time      Priority        Waiting Time    Turnaround Time
1       10              2               5               15
2       5               1               0               5
3       8               4               21              29
4       6               3               15              21

Average Waiting Time: 10.25
Average Turnaround Time: 17.50
--------------------------------
Process exited after 14.25 seconds with return value 0
Press any key to continue . . .
```