Experiment 5

Lab Objective: Write a C Program to Implement Round Robin CPU scheduling Algo.

```
#include <stdio.h>

void calculateWaitingTime(int processes[], int n, int burstTime[],
int waitingTime[], int quantum) {
    int remainingTime[n];
    for (int i = 0; i < n; i++) {
        remainingTime[i] = burstTime[i];
    }

    int currentTime = 0;
    int done;

    do {
        done = 1; // Assume all processes are done
        for (int i = 0; i < n; i++) {
            if (remainingTime[i] > 0) {
                done = 0; // There is a pending process
                if (remainingTime[i] > quantum) {
                    currentTime += quantum;
                    remainingTime[i] -= quantum;
                } else {
                    currentTime += remainingTime[i];
                    waitingTime[i] = currentTime - burstTime[i];
                    remainingTime[i] = 0;
                }
            }
        }
    } while (!done);
}

void calculateTurnAroundTime(int processes[], int n, int
burstTime[], int waitingTime[], int turnAroundTime[]) {
    for (int i = 0; i < n; i++) {
        turnAroundTime[i] = burstTime[i] + waitingTime[i];
    }
}
```

```
void calculateAverageTime(int processes[], int n, int burstTime[],
int quantum) {
    int waitingTime[n], turnAroundTime[n];
    int totalWaitingTime = 0, totalTurnAroundTime = 0;

    calculateWaitingTime(processes, n, burstTime, waitingTime,
quantum);
    calculateTurnAroundTime(processes, n, burstTime, waitingTime,
turnAroundTime);

    printf("Processes    Burst Time    Waiting Time    Turnaround
Time\n");

    for (int i = 0; i < n; i++) {
        totalWaitingTime += waitingTime[i];
        totalTurnAroundTime += turnAroundTime[i];
        printf("    %d           %d              %d              %d\
n",
               processes[i], burstTime[i], waitingTime[i],
turnAroundTime[i]);
    }

    printf("\nAverage Waiting Time = %.2f\n",
(float)totalWaitingTime / n);
    printf("Average Turnaround Time = %.2f\n",
(float)totalTurnAroundTime / n);
}

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

    int processes[n];
    int burstTime[n];
    int quantum;

    printf("Enter the burst time for each process:\n");
    for (int i = 0; i < n; i++) {
        processes[i] = i + 1;
        printf("Process %d: ", i + 1);
        scanf("%d", &burstTime[i]);
    }

    printf("Enter the time quantum: ");
```

```
    scanf("%d", &quantum);

    calculateAverageTime(processes, n, burstTime, quantum);

    return 0;
}
```

Output: