c) Round Robin

Aim: Write a C program to implement the various process scheduling mechanisms such as round robin scheduling

Program:

```
#include <stdio.h>

int main() {
    int n, tq, time = 0, completed = 0;
    int at[10], bt[10], rt[10];
    int wt[10], tat[10], ct[10], resp[10];
    int visited[10] = {0};
    float avg_wt = 0, avg_tat = 0, avg_resp = 0;

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

    for (int i = 0; i < n; i++) {
        printf("Enter Arrival Time and Burst Time for P%d: ", i + 1);
        scanf("%d %d", &at[i], &bt[i]);
        rt[i] = bt[i];
        resp[i] = -1;   // response time initialization
    }

    printf("Enter Time Quantum: ");
    scanf("%d", &tq);

    while (completed < n) {
        int executed = 0;

        for (int i = 0; i < n; i++) {
            if (rt[i] > 0 && at[i] <= time) {
                executed = 1;

                // First response
                if (resp[i] == -1)
                    resp[i] = time - at[i];
```

```
            if (rt[i] > tq) {
                time += tq;
                rt[i] -= tq;
            } else {
                time += rt[i];
                rt[i] = 0;
                ct[i] = time;
                completed++;
            }
        }
    }

    if (!executed)
        time++;  // CPU idle
}

printf("\nProcess\tAT\tBT\tWT\tTAT\tRT\n");

for (int i = 0; i < n; i++) {
    tat[i] = ct[i] - at[i];
    wt[i] = tat[i] - bt[i];

    avg_wt += wt[i];
    avg_tat += tat[i];
    avg_resp += resp[i];

    printf("P%d\t\t%d\t%d\t%d\t%d\t%d\n",
           i + 1, at[i], bt[i], wt[i], tat[i], resp[i]);
}

avg_wt /= n;
avg_tat /= n;
avg_resp /= n;

printf("\nAverage Waiting Time     = %.2f", avg_wt);
printf("\nAverage Turnaround Time  = %.2f", avg_tat);
```

```
    printf("\nAverage Response Time    = %.2f\n", avg_resp);

    return 0;
}
```

OUTPUT:

```
Enter number of processes: 3
Enter Arrival Time and Burst Time for P1: 0 3
Enter Arrival Time and Burst Time for P2: 1
4
Enter Arrival Time and Burst Time for P3: 3
5
Enter Time Quantum: 2

Process     AT    BT    WT    TAT   RT
P1          0     3     4     7     0
P2          1     4     4     8     1
P3          3     5     4     9     1

Average Waiting Time     = 4.00
Average Turnaround Time = 8.00
Average Response Time    = 0.67
```

c) Priority scheduling

Aim: Write a C program to implement the various process scheduling mechanisms such as Priority scheduling.

Program:

```
#include <stdio.h>

struct process {
    int pid;
    int burst_time;
    int priority;
    int waiting_time;
    int turnaround_time;
};

int main() {
    int n;
    struct process p[20];
    float avg_wt = 0, avg_tat = 0;

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

    for (int i = 0; i < n; i++) {
        p[i].pid = i + 1;
        printf("\nProcess %d", p[i].pid);
        printf("\nBurst Time: ");
        scanf("%d", &p[i].burst_time);
        printf("Priority: ");
        scanf("%d", &p[i].priority);
    }

    // Sort processes by priority (lower number = higher priority)
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (p[i].priority > p[j].priority) {
                struct process temp = p[i];
```

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

// Waiting time calculation
p[0].waiting_time = 0;
for (int i = 1; i < n; i++) {
    p[i].waiting_time = p[i - 1].waiting_time + p[i - 1].burst_time;
}

// Turnaround time calculation
for (int i = 0; i < n; i++) {
    p[i].turnaround_time = p[i].waiting_time + p[i].burst_time;
    avg_wt += p[i].waiting_time;
    avg_tat += p[i].turnaround_time;
}

avg_wt /= n;
avg_tat /= n;

printf("\n\nPID\tBT\tPR\tWT\tTAT");
for (int i = 0; i < n; i++) {
    printf("\n%d\t%d\t%d\t%d\t%d",
           p[i].pid,
           p[i].burst_time,
           p[i].priority,
           p[i].waiting_time,
           p[i].turnaround_time);
}

printf("\n\nAverage Waiting Time = %.2f", avg_wt);
printf("\nAverage Turnaround Time = %.2f\n", avg_tat);

return 0;
}
```

OUTPUT:

Enter number of processes: 4

Process 1
Burst Time: 10
Priority: 10

Process 2
Burst Time: 3
Priority: 20

Process 3
Burst Time: 1
Priority: 30

Process 4
Burst Time: 2
Priority: 40

PID	BT	PR	WT	TAT
1	10	10	0	10
2	3	20	10	13
3	1	30	13	14
4	2	40	14	16

Average Waiting Time = 9.25
Average Turnaround Time = 13.25