

8.ROUND ROBIN SCHEDULING

```
#include <stdio.h>

int main() {
    int n, tq, bt[20], wt[20], tat[20], rem_bt[20];
    int i, t = 0, count = 0;
    float avg_wt = 0, avg_tat = 0;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    printf("Enter burst times:\n");
    for (i = 0; i < n; i++) {
        printf("P%d: ", i + 1);
        scanf("%d", &bt[i]);
        rem_bt[i] = bt[i];
        wt[i] = tat[i] = 0;
    }
    printf("Enter Time Quantum: ");
    scanf("%d", &tq);
    while (1) {
        int done = 1;
        for (i = 0; i < n; i++) {
            if (rem_bt[i] > 0) {
                done = 0;
                if (rem_bt[i] > tq) {
                    t += tq;
                    rem_bt[i] -= tq;
                } else {
                    t += rem_bt[i];
                    wt[i] = t - bt[i];
                    rem_bt[i] = 0;
                }
            }
        }
    }
}
```

```

        tat[i] = t
    }
}
}
if (done == 1)
    break;
}
printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
for (i = 0; i < n; i++) {
    avg_wt += wt[i];
    avg_tat += tat[i];
    printf("P%d\tt%d\t\tt%d\t\tt%d\n", i + 1, bt[i], wt[i], tat[i]);
}
avg_wt /= n;
avg_tat /= n;
printf("\nAverage Waiting Time = %.2f", avg_wt);
printf("\nAverage Turnaround Time = %.2f\n", avg_tat);
return 0;
}

```