**ROUND ROBIN SCHEDULING**

#include <stdio.h>

int main() {

int i, n, time = 0, remain, flag = 0, tq;

int wait\_time = 0, turnaround\_time = 0;

int at[10], bt[10], rt[10]; // at: arrival time, bt: burst time, rt: remaining time

printf("Enter number of processes: ");

scanf("%d", &n);

remain = n;

// Input burst time (arrival time assumed 0)

for (i = 0; i < n; i++) {

printf("Enter burst time for Process P[%d]: ", i + 1);

scanf("%d", &bt[i]);

at[i] = 0; // All processes arrive at time 0

rt[i] = bt[i];

}

// Input time quantum

printf("Enter time quantum: ");

scanf("%d", &tq);

printf("\nProcess\tTurnaround Time\tWaiting Time\n");

// Round Robin Scheduling Loop

for (time = 0, i = 0; remain != 0;) {

if (rt[i] > 0 && rt[i] <= tq) {

time += rt[i];

rt[i] = 0;

flag = 1;

} else if (rt[i] > 0) {

rt[i] -= tq;

time += tq;

}

if (rt[i] == 0 && flag == 1) {

remain--;

int wt = time - at[i] - bt[i];

int tat = time - at[i];

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

wait\_time += wt;

turnaround\_time += tat;

flag = 0;

}

i = (i + 1) % n; // Rotate through processes

}

printf("\nAverage Waiting Time: %.2f", (float)wait\_time / n);

printf("\nAverage Turnaround Time: %.2f\n", (float)turnaround\_time / n);

return 0;

}