ROUND ROBIN SCHEDULING

#include<stdio.h>

#include<stdlib.h>

struct rr {

int pno, btime, sbtime, wtime, lst;

} p[10];

int main() {

int pp = -1, ts, flag, count, ptm = 0, i, n, twt = 0, totttime = 0;

printf("\nRound Robin Scheduling\n");

printf("Enter the number of processes: ");

scanf("%d", &n);

printf("Enter the time slice: ");

scanf("%d", &ts);

printf("Enter the burst time\n");

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

printf("Process %d: ", i + 1);

scanf("%d", &p[i].btime);

p[i].wtime = p[i].lst = 0;

p[i].pno = i + 1;

p[i].sbtime = p[i].btime;

}

printf("Scheduling....\n");

do {

flag = 0;

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

count = p[i].btime;

if(count > 0) {

flag = -1;

count = (count >= ts) ? ts : count;

printf("\nProcess %d from %d to ", p[i].pno, ptm);

ptm += count;

printf("%d", ptm);

p[i].btime -= count;

if(pp != i) {

pp = i;

p[i].wtime += ptm - p[i].lst - count;

p[i].lst = ptm;

}

}

}

} while(flag);

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

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

printf("%d\t%d\t\t%d\n", p[i].pno, p[i].sbtime, p[i].wtime);

twt += p[i].wtime;

totttime += p[i].sbtime;

}

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

printf("Average Turnaround Time: %.2f\n", (float)(twt + totttime) / n);

return 0;

}

OUTPUT:

