CPU Scheduling - Priority Scheduling

#Include <stdio.h>

int main() {

int n, i, j, temp;

int pid[20], burst[20], priority[20], wait[20], tat[20];

float avg\_wait = 0, avg\_tat = 0;

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

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

printf("Enter burst time for process %d: ", i + 1);

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

printf("Enter priority for process %d (lower value = higher priority): ", i + 1);

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

pid[i] = i + 1;

}

for (i = 0; i < n - 1; i++) {

for (j = i + 1; j < n; j++) {

if (priority[i] > priority[j]) {

temp = priority[i];

priority[i] = priority[j];

priority[j] = temp;

temp = burst[i];

burst[i] = burst[j];

burst[j] = temp;

temp = pid[i];

pid[i] = pid[j];

pid[j] = temp;

}

}

}

wait[0] = 0;

for (i = 1; i < n; i++) {

wait[i] = wait[i - 1] + burst[i - 1];

}

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

tat[i] = wait[i] + burst[i];

avg\_wait += wait[i];

avg\_tat += tat[i];

}

printf("\nProcess\tBurst\tPriority\tWaiting\tTurnaround\n");

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

printf("P%d\t%d\t%d\t\t%d\t%d\n", pid[i], burst[i], priority[i], wait[i], tat[i]);

}

printf("\nAverage Waiting Time = %.2f", avg\_wait / n);

printf("\nAverage Turnaround Time = %.2f\n", avg\_tat / n);

return 0;

}