**PRIORITY SCHEDULING**

#include <stdio.h>

int main() {

int n, i, j;

int burst[20], priority[20], process[20];

int waiting[20], turnaround[20];

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

// Input number of processes

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

scanf("%d", &n);

// Input burst time and priority for each process

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

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

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

process[i] = i + 1;

}

// Sort by priority (ascending: 1 is highest priority)

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

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

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

// Swap priority

int temp = priority[i];

priority[i] = priority[j];

priority[j] = temp;

// Swap burst time

temp = burst[i];

burst[i] = burst[j];

burst[j] = temp;

// Swap process ID

temp = process[i];

process[i] = process[j];

process[j] = temp;

}

}

}

// First process waiting time is 0

waiting[0] = 0;

// Calculate waiting time

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

waiting[i] = 0;

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

waiting[i] += burst[j];

}

}

// Calculate turnaround time

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

turnaround[i] = burst[i] + waiting[i];

avg\_wait += waiting[i];

avg\_turnaround += turnaround[i];

}

avg\_wait /= n;

avg\_turnaround /= n;

// Print results

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

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

printf("P[%d]\t%d\t\t%d\t\t%d\t\t%d\n", process[i], burst[i], priority[i], waiting[i], turnaround[i]);

}

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

printf("\nAverage Turnaround Time: %.2f\n", avg\_turnaround);

return 0;

}