OPERATING SYSTEMS
PROGRAM-2

Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time.

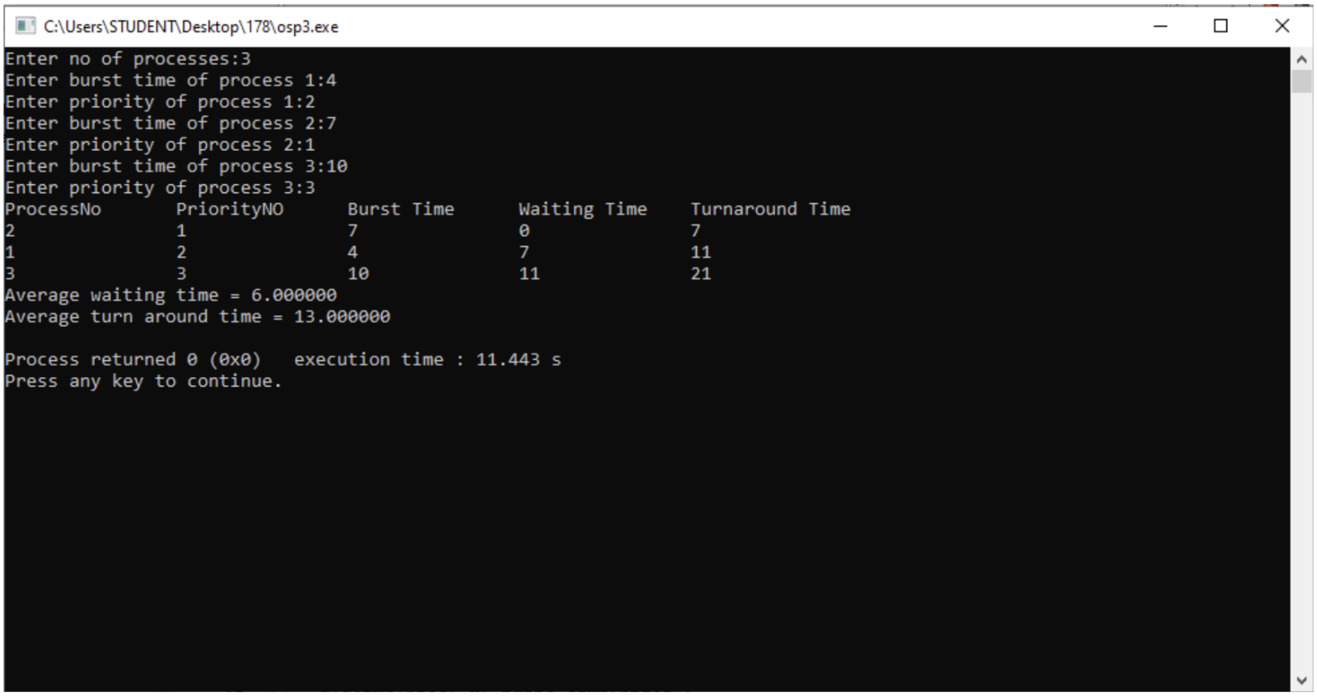
□ Priority (pre-emptive & Non-pre-emptive)

□Round Robin (Experiment with different quantum sizes for RR algorithm)

- PRIORITY

```
#include<stdio.h>

int waitingtime(int proc[], int n, int burst_time[], int wait_time[]){
  int i;
  wait_time[0]=0;
  for(i=1;i<n;i++){
   wait_time[i]=burst_time[i-1]+wait_time[i-1];
 }
}
int turnaroundtime(int proc[], int n, int burst_time[], int wait_time[], int tat[]) {
 for (int i = 0; i < n; i++) {
  tat[i] = burst_time[i] + wait_time[i];
 }
}

int avgtime(int proc[], int n, int burst_time[],int prior[]) {
 int wait_time[n], tat[n], total_wt = 0, total_tat = 0,temp1,temp2,temp3;

 for(int i=0;i<n;i++){
  for(int j=i+1;j<n;j++){
    if(prior[i]>prior[j]){
      temp1=burst_time[i];
      burst_time[i]=burst_time[j];
      burst_time[j]=temp1;
```

```
            temp2=prior[i];
            prior[i]=prior[j];
            prior[j]=temp2;
            temp3=proc[i];
            proc[i]=proc[j];
            proc[j]=temp3;
        }
    }
  }

  waitingtime(proc, n, burst_time, wait_time);
  turnaroundtime(proc, n, burst_time, wait_time, tat);

  printf("ProcessNo\tPriorityNO\tBurst Time\tWaiting Time\tTurnaround Time\n");

  for (int i = 0; i < n; i++) {
    total_wt = total_wt + wait_time[i];
    total_tat = total_tat + tat[i];
    printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", proc[i],prior[i],burst_time[i], wait_time[i], tat[i]);
  }

  printf("Average waiting time = %f\n", (float)total_wt / (float)n);
  printf("Average turn around time = %f\n", (float)total_tat / (float)n);
}

int main() {
  int i,proc[10],n,burst_time[10],prior[10];
  printf("Enter no of processes:");
  scanf("%d",&n);
  for(i=0;i<n;i++){
    printf("Enter burst time of process %d:",i+1);
    scanf("%d",&burst_time[i]);
    printf("Enter priority of process %d:",i+1);
    scanf("%d",&prior[i]);
    proc[i]=i+1;
  }
  avgtime(proc, n, burst_time,prior);

  return 0;
}
```

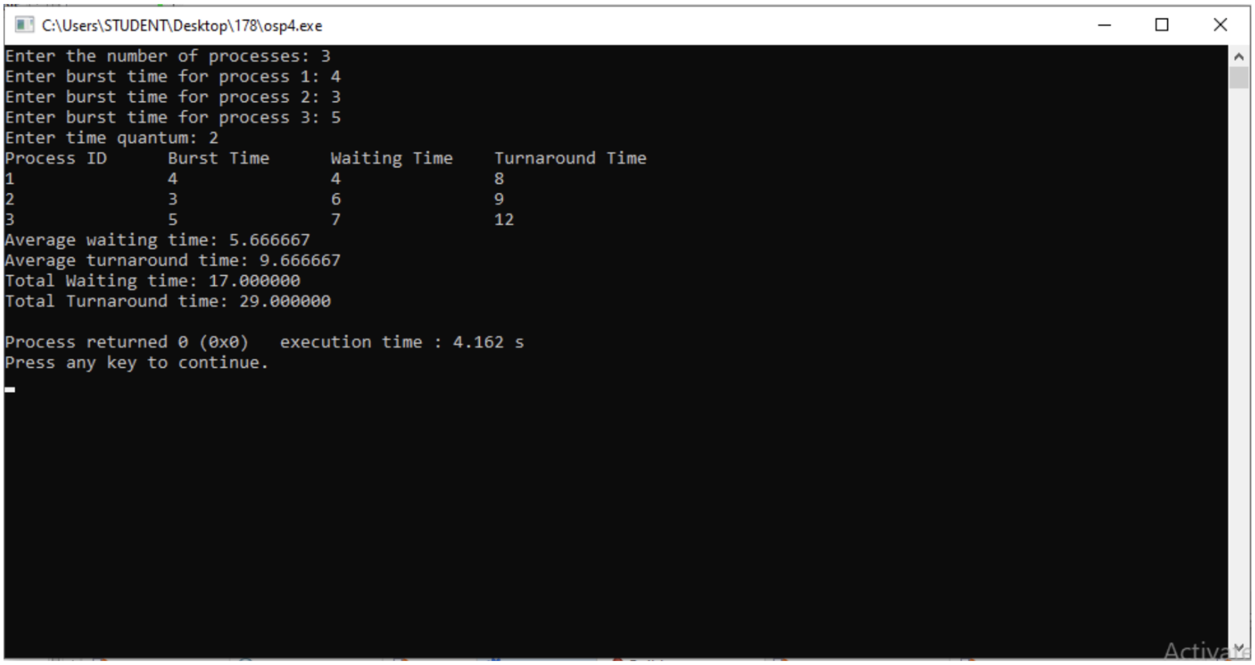


```
Enter no of processes:3
Enter burst time of process 1:4
Enter priority of process 1:2
Enter burst time of process 2:7
Enter priority of process 2:1
Enter burst time of process 3:10
Enter priority of process 3:3
ProcessNo       PriorityNO      Burst Time      Waiting Time    Turnaround Time
2               1               7               0               7
1               2               4               7               11
3               3               10              11              21
Average waiting time = 6.000000
Average turn around time = 13.000000

Process returned 0 (0x0)   execution time : 11.443 s
Press any key to continue.
```

- ROUND ROBIN

```
#include <stdio.h>

int main() {
  int n, quantum, i, j;
  printf("Enter the number of processes: ");
  scanf("%d", &n);

  int burst_time[n], remaining_time[n], waiting_time[n], turnaround_time[n];

  for (i = 0; i < n; i++) {
    printf("Enter burst time for process %d: ", i + 1);
    scanf("%d", &burst_time[i]);
    remaining_time[i] = burst_time[i];
  }

  printf("Enter time quantum: ");
  scanf("%d", &quantum);

  int time = 0, done = 0;
  while (done != n) {
    for (i = 0; i < n; i++) {
      if (remaining_time[i] > 0) {
```

```
      if (remaining_time[i] > quantum) {
        time += quantum;
        remaining_time[i] -= quantum;
      } else {
        time += remaining_time[i];
        waiting_time[i] = time - burst_time[i];
        remaining_time[i] = 0;
        turnaround_time[i] = time;
        done++;
      }
    }
  }
}

float total_waiting_time = 0, total_turnaround_time = 0;
for (i = 0; i < n; i++) {
  total_waiting_time += waiting_time[i];
  total_turnaround_time += turnaround_time[i];
}

printf("Process ID\tBurst Time\tWaiting Time\tTurnaround Time\n");
for (i = 0; i < n; i++) {
  printf("%d\t\t%d\t\t%d\t\t%d\n", i + 1, burst_time[i], waiting_time[i], turnaround_time[i]);
}

printf("Average waiting time: %f\n", total_waiting_time / n);
printf("Average turnaround time: %f\n", total_turnaround_time / n);
printf("Total Waiting time: %f\n", total_waiting_time);
printf("Total Turnaround time: %f\n", total_turnaround_time);

return 0; }
```



```
Enter the number of processes: 3
Enter burst time for process 1: 4
Enter burst time for process 2: 3
Enter burst time for process 3: 5
Enter time quantum: 2
Process ID      Burst Time      Waiting Time    Turnaround Time
1               4               4               8
2               3               6               9
3               5               7               12
Average waiting time: 5.666667
Average turnaround time: 9.666667
Total Waiting time: 17.000000
Total Turnaround time: 29.000000

Process returned 0 (0x0)   execution time : 4.162 s
Press any key to continue.
```