PROGRAM NUMBER :2

AIM:

Write a C program to simulate then non-pre-emptive CPU scheduling algorithms for finding turnaround time and waiting time.

a) Round Robin (pre-emptive)

b) Priority

PROGRAM

1.Round Robin (pre-emptive)

```
ng@ng-TravelMate-5742:~/system$ cat pre-emptive\ round\ robin.c
#include<stdio.h>
int main()
{
  int count,j,n,time,remain,flag=0,time_quantum;
  int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
  printf("Enter Total Process:\t ");
  scanf("%d",&n);
  remain=n;
  for(count=0;count<n;count++)
  {
    printf("Enter Arrival Time and Burst Time for P%d :",count+1);
    scanf("%d",&at[count]);
    scanf("%d",&bt[count]);
    rt[count]=bt[count];
  }
  printf("Enter Time Quantum:\t");
  scanf("%d",&time_quantum);
  printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
  for(time=0,count=0;remain!=0;)
  {
    if(rt[count]<=time_quantum && rt[count]>0)
    {
      time+=rt[count];
      rt[count]=0;
      flag=1;
    }
    else if(rt[count]>0)
    {
      rt[count]-=time_quantum;
      time+=time_quantum;
    }
    if(rt[count]==0 && flag==1)
    {
      remain--;
      printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-
      wait_time+=time-at[count]-bt[count];
      turnaround_time+=time-at[count];
```

```
      flag=0;
    }
    if(count==n-1)
      count=0;
    else if(at[count+1]<=time)
      count++;
    else
      count=0;
  }
  printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
  printf("Avg Turnaround Time = %f",turnaround_time*1.0/n);

  return 0;
}

ng@ng-TravelMate-5742:~/system$
```

OUTPUT

```
ng@ng-TravelMate-5742:~/system$ gcc pre-emptive\ round\ robin.c
ng@ng-TravelMate-5742:~/system$ ./a.out
Enter Total Process:     4
Enter Arrival Time and Burst Time for P1 :0 3
Enter Arrival Time and Burst Time for P2 :1 6
Enter Arrival Time and Burst Time for P3 :2 6
Enter Arrival Time and Burst Time for P4 :3 8
Enter Time Quantum:     4


Process |Turnaround Time|Waiting Time

P[1]    |       3       |       0
P[2]    |       16      |       10
P[3]    |       17      |       11
P[4]    |       20      |       12

Average Waiting Time= 8.250000
Avg Turnaround Time = 14.000000
ng@ng-TravelMate-5742:~/system$
```

PROGRAM

b. Priority (non - preemptive)

```
ng@ng-TravelMate-5742:~/system$ cat non-preemptive\ \ priority.c
#include<stdio.h>
int main()
{
      int burst_time[20], process[20], waiting_time[20], turnaround_time[20]
      int i, j, limit, sum = 0, position, temp;
      float average_wait_time, average_turnaround_time;
      printf("Enter Total Number of Processes:\t");
      scanf("%d", &limit);
      printf("\nEnter Burst Time and Priority For %d Processes\n", limit);
      for(i = 0; i < limit; i++)
      {
            printf("\nProcess[%d]\n", i + 1);
            printf("Process Burst Time:\t");
            scanf("%d", &burst_time[i]);
            printf("Process Priority:\t");
            scanf("%d", &priority[i]);
            process[i] = i + 1;
      }
      for(i = 0; i < limit; i++)
      {
            position = i;
            for(j = i + 1; j < limit; j++)
            {
                  if(priority[j] < priority[position])
                  {
                        position = j;
                  }
            }
            temp = priority[i];
            priority[i] = priority[position];
            priority[position] = temp;
            temp = burst_time[i];
            burst_time[i] = burst_time[position];
            burst_time[position] = temp;
            temp = process[i];
            process[i] = process[position];
            process[position] = temp;
```

```
}
waiting_time[0] = 0;
for(i = 1; i < limit; i++)
{
      waiting_time[i] = 0;
      for(j = 0; j < i; j++)
      {
            waiting_time[i] = waiting_time[i] + burst_time[j];
      }
      sum = sum + waiting_time[i];
}
average_wait_time = sum / limit;
sum = 0;
printf("\nProcess ID\t\tBurst Time\t Waiting Time\t Turnaround Time\n");
for(i = 0; i < limit; i++)
{
      turnaround_time[i] = burst_time[i] + waiting_time[i];
      sum = sum + turnaround_time[i];
      printf("\nProcess[%d]\t\t%d\t\t %d\t\t %d\n", process[i], burst_time[i
}
average_turnaround_time = sum / limit;
printf("\nAverage Waiting Time:\t%f", average_wait_time);
printf("\nAverage Turnaround Time:\t%f\n", average_turnaround_time);
return 0;


-TravelMate-5742:~/system$
```

OUTPUT

```
Enter Total Number of Processes:        3

Enter Burst Time and Priority For 3 Processes

Process[1]
Process Burst Time:     15
Process Priority:       3

Process[2]
Process Burst Time:     10
Process Priority:       2

Process[3]
Process Burst Time:     90
Process Priority:       1

Process ID              Burst Time      Waiting Time    Turnaround Time

Process[3]              90              0               90

Process[2]              10              90              100

Process[1]              15              100             115

Average Waiting Time:   63.000000
Average Turnaround Time:        101.000000
ng@ng-TravelMate-5742:~/system$
```

RESULT

Program is executed successfully and output is obtained.

BY NIVEA GIGEN
S5 C
CHN18CS092