PAGE NO:
DATE: 28/6/23

Week-3

Write a C program to execute round robin and non preemptive priority scheduling.

```
#include <stdio.h>
int at[20], cput[20];
void main()
{
    int n, i, choice, tq, v;
    printf("Enter the number of process \n");
    scanf("%d", &n);
    printf("Enter the arrival time and cpu time for each process respectively \n");
    for(i=0; i<n; i++) {
        scanf("%d %d", &at[i], &cput[i]);
    }
    printf("\n Menu \n\n1. Round Robin \n2. priority (non-preemptive) \n3. exit \n");
    while(1) {
    scanf("%d", &choice);
    switch(choice) {
        case 1: printf("Enter the time quantum \n");
                scanf("%d", &tq);
                roundRobin(n, tq);
                break;
        case 2:
                Nonpreemptivepriority(n);
                break;
        case 3: exit(0);
        default: printf("Wrong choice \n");
    }
    }
}
```

```
void roundRobin(int n, int tq){
    int i, remaining_time[20], wt[20], tat[20],
    completed = 0, time = 0;
    float awt = 0, atat = 0;

    for(i=0; i<n; i++){
        remaining_time[i] = cput[i];
    }

    while(completed <n){
    for(int i=0; i<n; i++){
        if(remaining_time[i]>0 && at[i]<=time){
        if(remaining_time[i] <=tq){
            time += remaining_time[i];
            remaining_time[i] = 0;
            completed ++;
            tat[i] = time - at[i];
            wt[i] = tat[i] - cput[i];
        }
        else{
            time += tq;
            remaining_time[i] = tq;
        }
    }
    }

    for(i=0; i<n; i++){
        atat += tat[i];
        awt += wt[i];
    }
    atat /= n;
    awt /= n;
```

```
printf("\nProcess\t cputime\t arrivaltime\t turnaround
        time\t waitingtime\n");
for(int i=0 ; i<n ; i++){
    printf("%d \t %d \t \t %d \t \t %d \t \t %d\n",
            i, cput[i], at[i], tat[i], wt[i]);
    }
printf("Average Turnaround time : %.2f\n", atat);
printf("Average waiting time : %.2f\n", awt);
}

Void Nonpre priority(int n){
    int priority[20], wt[20], tat[20], hp=0,
    cmpt[20], cput1[20], sum=0, i, sum_burst_time=0;
    float awt=0, atat=0;

printf("Enter the priorities of processes\n");
  for(int i=0 ; i<n ; i++){
    printf("process %d : ", i);
    scanf("%d", &priority[i]);
  }
for(i=0 ; i<n ; i++){
    cput1[i] = cput[i];
    sum_burst_time += cput[i];
  }
printf("\tprocess \t turnaround time \t waiting time\n");
  cput1[9] = -1;
  while(sum < sum_burst_time){
        hp = 9;
    for(i=0; i< i++){
      if(at[i] <= sum && cput1[i] > 0 &&
          priority[i] > priority[hp])
            hp = i;
    }
```

```
printf("\t p[%d] \t\t %d \t\t\t %d \n", hp[...],
        sum + cput[hp] - at[hp], sum - at[hp]);
    awt += sum + cput[hp] - at[hp];
    atat += sum - at[hp];
    sum += cput[hp];
    cput[hp] = 0;
    }
awt = awt / n;
atat = atat / n;
printf("\n Average waiting time ... %f", awt);
printf("\n Average Turnaround time ... %f\n", atat);
}
```

Output :

Enter the number of process

5

Enter the arrival time & cpu time for each process respectively

0 5

1 3

2 1

3 2

4 2

Menu

1. Round Robin
2. Priority (Non preemptive)
3. Exit.

1.

Enter the time quantum

2

Process	Cputime	Arrival Time	Turnaround time	Waiting time
0	5	0	14	9
1	3	1	11	8
2	1	2	3	2
3	2	3	4	2
4	3	4	9	6

Average turn around time : 8.20
Average waiting time : 5.40

2
Enter the priorities of processes
process 0 : 3
process 1 : 2
process 2 : 1
process 3 : 4
process 4 : 3

Process	turn around time	Waiting time
p[0]	5	0
p[3]	4	2
p[4]	6	3
p[1]	12	9
p[2]	12	11

Average waiting Time : 5.00
Average Turn around time : 7.80

PAGE NO :
DATE :

Waiting time

9
8
2
2
6

Gantt chart

Round Robin

| P1 | P2 | P3 | P1 | P4 | P5 | P2 | P1 | P5 |
0 2 4 5 7 9 11 12 13 14

P2(3) P3(1) P1(3)
P3(1) P1(2) P4(2)
P1(3) P4(2) P5(3)
P5(3) P2(1)
P2(1)

P4(2) P5(3) P2(1) P1(1)
P5(3) P2(1) P1(1) P5(1)
P1(1) P5(1)

P5(1)

Priority Non Preemptive.

| P1 | P4 | P5 | P2 | P3 |
0 5 7 10 13 14

P2(3) P2(3) P2(3) P3(1)
P3(1) P3(1) P3(1)
P4(2) P5(3)
P5(3)

6/10

19/7/23

```
Enter the number of processes: 3

Enter the process ID: 1
Enter the burst time: 3
Enter the priority: 2

Enter the process ID: 2
Enter the burst time: 4
Enter the priority: 1

Enter the process ID: 3
Enter the burst time: 2
Enter the priority: 3

Process ID      Burst Time      Priority        Waiting Time    Turnaround Time
2               4               1               0               4
1               3               2               4               7
3               2               3               7               9

Average Waiting Time = 3.666667
Average Turnaround Time = 6.666667
```

```
Enter number of processes: 3

 Enter the Arrival and Burst time of the Process[1]: 2 12

 Enter the Arrival and Burst time of the Process[2]: 0 3

 Enter the Arrival and Burst time of the Process[3]: 7 8
Enter the time quantum for the process:         2

 Process No              Burst Time              TAT             Waiting Time
Process No[2]            3                       7                       4
Process No[3]            8                       14                      6
Process No[1]            12                      21                      9
 Average Turn Around Time:      6.333333
 Average Waiting Time:  14.000000
```