Lab 3: Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time.
→ Priority (pre-emptive or Non-pre-emptive)
→Round Robin (Experiment with different quantum sizes for RR algorithm)

store
67

28/6/'23.

Q2. Write a C program to simulate the following CPU Scheduling algorithm to find turnaround time & waiting time
- Priority (Non pre-emptive)
- Round Robin.

```
#include <stdio.h>
int n, at[20], cput[20];

void roundRobin()
{
  int TQ, i;
  int remaining_time[n], wt[n], tat[n];
  int completed_processes = 0, current_time = 0;

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

  printf("Enter the time quantum : ");
  scanf("%d", &TQ);
  //RoundRobin Scheduling
  while (completed_processes < n)
  {
    for (i = 0; i < n; i++)
    {
      if (remaining_time[i] > 0 && at[i] <=
          current_time)
      {
        if (remaining_time[i] <= TQ)
        {
          current_time += remaining_time[i];
```

```
            remaining_time[i] = 0;
            completed_processes++;

            tat[i] = current_time - at[i];
            wt[i] = tat[i] - cput[i];
        }
        else
        {
            current_time += TQ;
            remaining_time[i] -= TQ;
        }
    }
}

float avgtat = 0, avgwt = 0;

for(i = 0; i < n; i++)
{
    avgtat += tat[i];
    avgwt += wt[i];
}

avgtat /= n;
avgwt /= n;

printf("\nProcess\t CPU Time\t Arrival Time\t
Turnaround Time\t Waiting Time\n");

for(i = 0; i < n; i++)
{   printf("%d\t %d\t %d\t %d\t %d\t %d\n",
        i, cput[i], at[i], tat[i], wt[i]);
```

```
    printf("Average Turnaround Time : %.2f\n",
            avgtat);
    printf("Average Waiting Time: %.2f \n", avgwt);
}

void NonPrePriority()
{
  int completion_time[n], tat[n], wt[n],
  priority[n], cputd[n];
  float avgwt=0, avgtat=0, sum_cpu_time=0;
  int time=0, i, j, curr_hp, temp2;

  for(i=0; i<n; i++)
  {
    printf("\n Enter Priority for the Process %d :", i);
    scanf("%d", &priority[i]);
  }

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

  cputd[9] = -1;
  while(time < sum_cpu_time)
  {
    curr_hp = 9;
    for(i=0; i<n; i++)
    {
      if(at[i] <= time && cputd[i] > 0 &&
          priority[curr_hp])
```

```
        curr_hp = i;
    }
    printf("P[%d]\t|\t%d\n", curr_hp, time
            + cputd[curr_hp] - at[curr_hp],
            time - at[curr_hp]);
    avgwt += time + cputd[curr_hp] - at[curr_hp];
    avgtat += time - at[curr_hp];
    time += cput[curr_hp];
    cputd[curr_hp] = 0;
}

avgwt = avgwt/n;
avgtat = avgtat/n;

    printf("\nAverage Waiting Time : %f", avgit);
    printf("\nAverage TAT time : %f", avgtat);
}


void main()
{
  int i, ch;
  printf("\nEnter the number of processes : ");
  scanf("%d", &n);
  for(i=0; i<n; i++)
  {
    printf("\nEnter Arrival Time and CPU Time
        for the Process %d : ", i);
    scanf("%d %d", &at[i], &cput[i]);
}
```

```
while (1)
{
  printf ("MENU \n");
  printf ("1. Round Robin \n 2. Priority \n 3. Exit \n");
  scanf ("%d", &ch);

  Switch (ch)
  {
    case 1 : roundRobin();
             break;
    case 2 : NonPrePriority();
             break;
    case 3 : exit(0);
    default : printf ("\n Wrong Choice !! Try Again ");
  }
}
}
```

Output :-

Enter the number of processes : 5

Enter Arrival Time and CPU time for the Process ~~0 : 0~~ :

0 5
1 3
2 1
3 2
4 3

MENU.
1. Round Robin
2. Priority
3. Exit
1
Enter the time quantum : 2

Process	CPU Time	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 TAT : 8.20
Average WT : 5.40

MENU.
1. Round Robin
2. Priority
3. Exit
2

Enter Priority for the Process
3
2
1
4
3

store
67

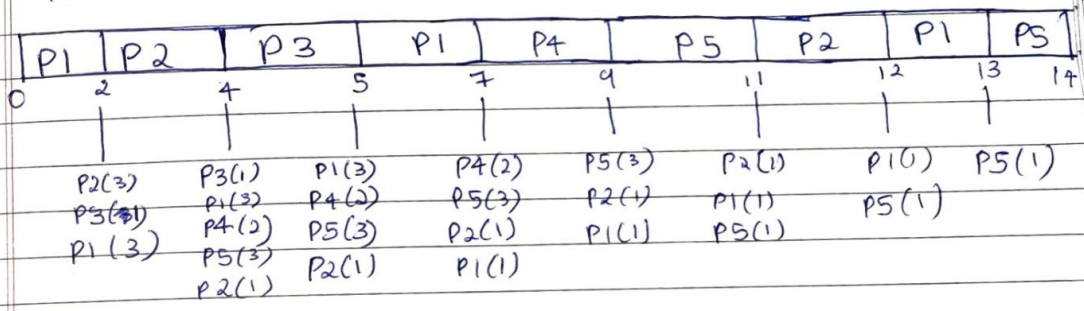
P[0]	5	0
P[3]	4	2
P[4]	6	3
P[1]	12	9
P[2]	12	11

Average Waiting Time : 5.000000
Average Turn Around Time : 7.800000

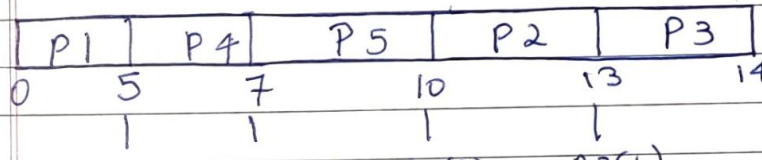
3

Gantt charts

Round Robin :

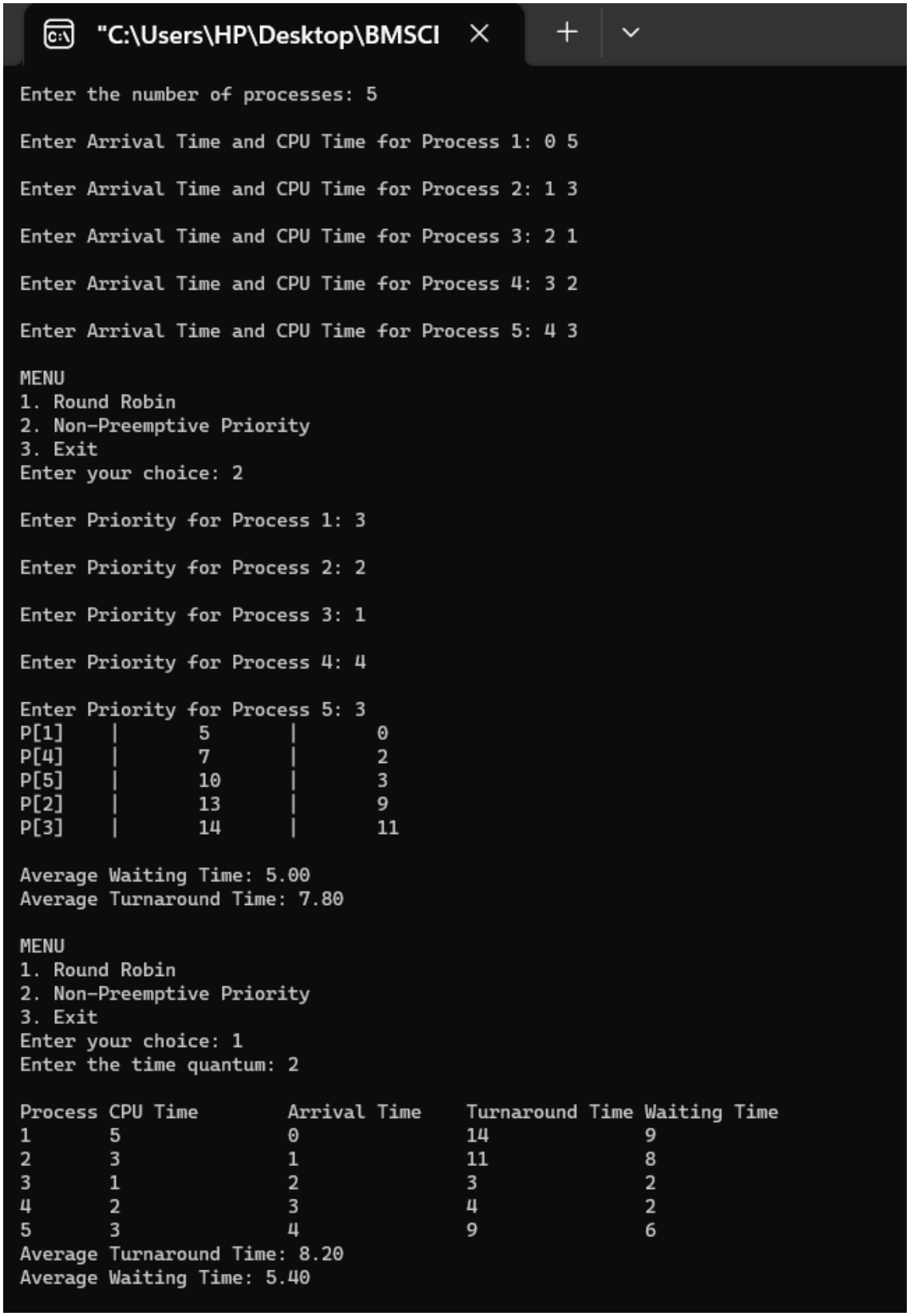


Priority Non-Preemptive :-



29/6/2022

OUTPUT:


"C:\Users\HP\Desktop\BMSCI
Enter the number of processes: 5
Enter Arrival Time and CPU Time for Process 1: 0 5
Enter Arrival Time and CPU Time for Process 2: 1 3
Enter Arrival Time and CPU Time for Process 3: 2 1
Enter Arrival Time and CPU Time for Process 4: 3 2
Enter Arrival Time and CPU Time for Process 5: 4 3
MENU
1. Round Robin
2. Non-Preemptive Priority
3. Exit
Enter your choice: 2
Enter Priority for Process 1: 3
Enter Priority for Process 2: 2
Enter Priority for Process 3: 1
Enter Priority for Process 4: 4
Enter Priority for Process 5: 3
P[1] | 5 | 0
P[4] | 7 | 2
P[5] | 10 | 3
P[2] | 13 | 9
P[3] | 14 | 11
Average Waiting Time: 5.00
Average Turnaround Time: 7.80
MENU
1. Round Robin
2. Non-Preemptive Priority
3. Exit
Enter your choice: 1
Enter the time quantum: 2
Process CPU Time Arrival Time Turnaround Time Waiting Time
1 5 0 14 9
2 3 1 11 8
3 1 2 3 2
4 2 3 4 2
5 3 4 9 6
Average Turnaround Time: 8.20
Average Waiting Time: 5.40