Ex. No.: 6c)
Date: 6.3.25.

PRIORITY SCHEDULING

Aim:

To implement priority scheduling technique

Algorithm:

1. Get the number of processes from the user.
2. Read the process name, burst time and priority of process.
3. Sort based on burst time of all processes in ascending order based priority 4. Calculate the total waiting time and total turnaround time for each process 5. Display the process name & burst time for each process.
6. Display the total waiting time, average waiting time, turnaround time

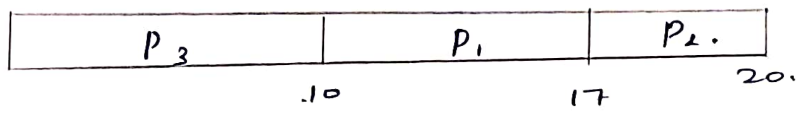
Program Code:

```
# include<stdio.h>
int main ().
{
    int n;
    printf ("Enter the no. of processes: ");
    Scanf ("%d", &n);
    int at[n], bt[n], priority [n], wt[n], tat[n] ; ct[n], pro[n]);
   for (int i=0; i<n; i++).
   {
        pro[i] = i+1;
        printf (" process  %d  Arrival time : ", i+1);
        scanf (" %d", & at[i]);
        printf (" process %d  Burst time : ", i+1);
        Scanf (" %d", & priority [i]);
        Scanf (" %d", & priority [i]);
   }
   for (int i=0; i<n; i++).
   {
        for (int j=0; j<n-i-1; j++)
        {
            if (at[j] > at[j+1])
            {
                int temp;
                temp = at [j]; 41
                ~~at[j]~~ = at [j+1];
                at [j+1] = temp;
                temp = priority [j];
                priority [j] = priority [j+1];
                priority [j+1] = temp;
                temp = pro[j];
                pro[j] = pro[j+1];
                pro [j+1] = temp;
            }
        }
    }
int time = 0, comp = 0; float total_tat = 0, total_wt = 0;
while (comp < n){
        int start = comp , end = comp;
        while ( end < n && at [end] <= time)

            end ++
    for (int i = start ; i < end - 1; i++){
        for (int j = start; i < end - i - 1; j++){
            if (priority [j] > priority [j+1]){
                int temp;
                temp = at[j];
                at[j] = at [j+1];
                at[j+1] = temp;
                temp = bt[j] ; bt [j] = bt [j+1];
                bt [j+1] = temp;
                temp = priority [j]; priority [j] = priority [j+1].
                priority [j+1] = temp;
                temp = pro[j] ; pro[j] = pro [j+1];
                pro[j+1] = temp;
    }   }   }
    time = (time < at [comp]) ? at [comp] : time;
    at [comp ] = time + bt [comp];
    tat [comp] = ct [comp]  - at [comp];
    wt [comp] = tat [comp] - bt [comp];
    time = ct [comp];
    comp ++)
}.
```

```
printf ("process Arrival time  Burst time  turn around time
        waiting time , priority);
for ( int i=0;  i<n; i++){
    printf(" %d    %d  %d  %d %d\n",  pro[i],  at[i], bt[i],
                        tat [i] ,  wt [i] ,  priority (i]);
  total _ tat  + =  tat [i];
  total_ wt  + =  wt [i];
}.
printf ( " Average  turn around time : %.2F ", total tat \n);
printf (" Average  ~~t~~waiting  time : %.2F  ", total wait \n);
return 0;
}.
```

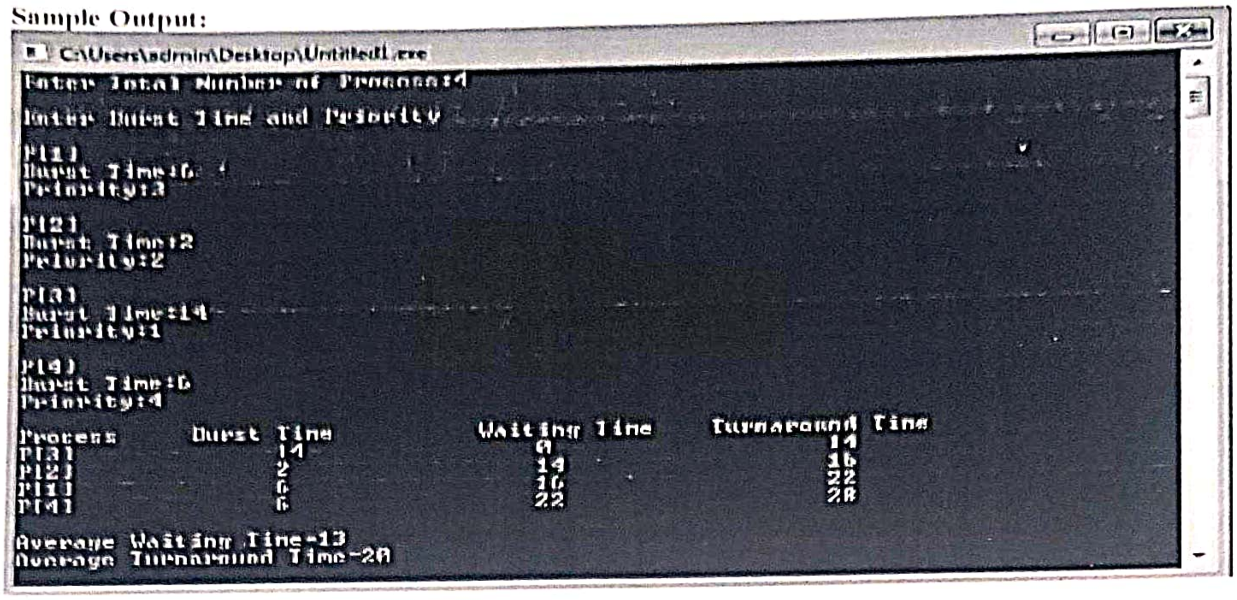
Gantt chart



tabulation.

Process	Bt (ms)	Priority	At (ms)	ct (ms)	TAT = Ct - At (ms)	Wt = TAT - BT (ms).
1	7	2	0	17	17	10
2	3	3	0	20	20	17
3	10	1	0	10	10	0

Sample Output:



```
C:\Users\admin\Desktop\Untitled1.exe
Enter Total Number of Process:4

Enter Burst Time and Priority

P[1]
Burst Time:6
Priority:3

P[2]
Burst Time:2
Priority:2

P[3]
Burst Time:14
Priority:1

P[4]
Burst Time:6
Priority:4

Process     Burst Time          Waiting Time     Turnaround Time
P[3]             14                 0                  14
P[2]             2                  14                 16
P[1]             6                  16                 22
P[4]             6                  22                 28

Average Waiting Time=13
Average Turnaround Time=20
```

```
Enter the no. of process :3.
Enter the process 1 Burst time: 7
Enter the process 2 Burst time: 3
Enter the process 3 Burst time: 10.
Enter the priority of process 1: 2
Enter the priority of process 2: 3
Enter the priority of process 3: 1
Process   Burst time   Priority   turnaround time   waiting time.
1            10           1            10               0
2            7            2            17               10
3            3            3            20               17
Average turn around time: 15.66.
Average waiting time: 9.00.
```

Result:

Thus the implementation of priority cpu scheduling has been successfully executed.

[illegible]