Ex. No.: 6c)
Date: 6/3/25

PRIORITY SCHEDULING

Aim:

To implement priority scheduling technique

Algorithm:

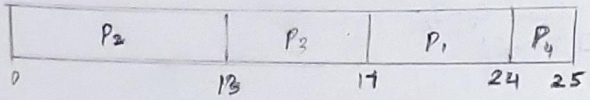
1. Get the number of processes from the user.
2. Read the process name, burst time and priority of process.
3. Sort based on burst time of all processes in ascending order based priority 4.
Calculate the total waiting time and total turnaround time for each process 5.
Display the process name & burst 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 process: ");
 scanf("%d",&n);
 int at[n], bt[n], priority[n], wt[n], tat[n], ct[n], process[n];
 for(int i=0; i< n; i++)
 {
  process[i] = i+1;
  printf("Process %d Arrival time: ", i+1);
  scanf("%d", &at[i]);
 printf("Process %d Burst time: ", i+1);
  scanf("%d", &bt[i]);
  printf("Process %d Priority: ", i+1);
  scanf("%d", &priority[i]);
 }
 for(int i=0; i<n-1; i++)
  for(int j=0; j<n-i-1; j++)
  {
   if(at[j] > at[j+1])
   {
    int temp;
    temp = at[j];
    at[j] = at[j+1];
    at[j+1] = temp;
    temp = priority[j];
```

```
        priority[j+1] = temp;
        temp = process[j];
        process[j] = process[j+1];
        process[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; i++) {
        for (int j = start; j < 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 = process[j];
                process[j] = process[j+1];
                process[j+1] = temp;
            }
        }
    }
    time = (time < at[comp]) ? at[comp] : time;
    at[comp] = time + bt[comp];
    tat[comp] = ct[comp] - at[comp];
    ct[comp] = tat[comp] - bt[comp];
    time = ct[comp];
    comp++;
}
printf("Process  Arrival Time  Burst Time  Turnaround Time  Waiting Time  Priority");
for (int i = 0; i < n; i++) {
    printf("\n%d  %d  %d  %d  %d  %d", process[i], at[i],
           bt[i], tat[i], wt[i], priority[i]);
    total_tat += tat[i];
    total_wt += wt[i];
}
```

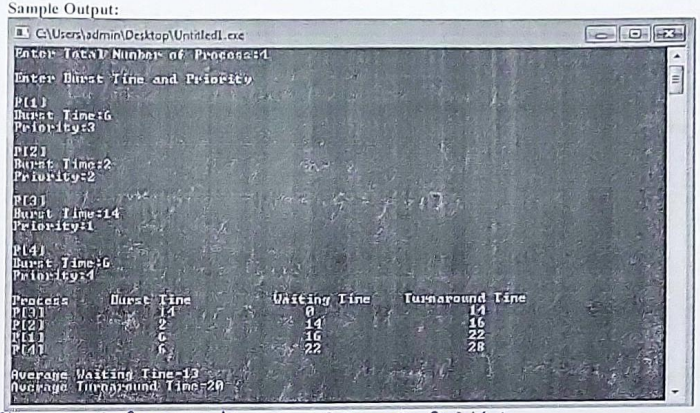
Gantt Chart:



Calculation:

Process	AT (ms)	BT (ms)	CT (ms)	Priority	TAT = CT - AT (ms)	WT = TAT - BT (ms)
1	0	7	24	3	24	17
2	0	13	13	1	13	0
3	0	4	17	2	17	13
4	0	1	25	4	25	24

Sample Output:



Enter no. of process: 4

Enter the burst time:
7
13
4
1

Enter the priority:
3
1
2
4

Process	Burst Time	Waiting time	Turn Around time
2	13	0	13
3	4	13	17
1	7	17	24
4	1	24	25

Avg. waiting time: 13.50 ms

Avg. turn around time: 19.75 ms

Result:

Thus the implementation of priority scheduling has been executed successfully.