Ex. No.: 6c)
Date: 7

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>
# include <stdlib.h>
int main(){
    int n;
    printf("Enter no. of process:\n");
    scanf("%d",& n);
    int bt[n], pr[n], ct[n], wt =0, tat[n], wt[n],
    at=0, awt =0

    printf("Enter the burst time");
    for(int i=0; i<n; i++)
        scanf("%d",&bt[i]);
    printf("Enter the priority of process:\n
    for(int i=0; i<n; i++)
        scanf("%d",&pr[i]);
    int sp[n],
for(int i=0; i<n; i++)
        printf("%d\n", wt[i]);
  for(int i=0; i<n; i++){
        atat = atat + tat[i];
        awt = awt + wt[i];
  printf("\n Average Turnaround time:
        \n Average wait time: %.2f\n"
        (float) atat \n, (float) awt \n);
```

Output:-

```
 Enter the no. of process:- 4
 Enter the bursttime:- 13
                       5
                       8
                       4

  Enter the priority of process:-
        3
        4
        2
        1

  Completion time: 22          Turn around Time:
                   9
                   30
                   4

  wait time: 9
             4
             22
             0

  Average Turnaround Time: 16.25 ms
   Average Wait Time: 8.75 ms
```

```
for(int i=0; i<n; i++)
      np[i] = p[i];
for(int i=0; i<n-1; i++){
      for(int j=0; j<n-1; j++){
            if(np[j+1] < np[j]){
                  int temp = np[j+1];
                  np[j+1] = np[j];
                  np[j] = temp;
            }
      }
}
int c=0;
for(int i=0; i<n; i++){
      for(int j=0; j<n; j++){
            if(np[i] == p[j]){
                  ct[j]=c+bt[j];
                  c=ct[j];
                  tat[j] = ct[j]-at;
                  wt[j] = tat[j]-bt[j];
            }
      }
}
printf("\n Completion time\n");
for(int i=0; i<n; i++)
      printf("%d \n", ct[i]);
printf("\n Turn around time \n");
for(int i=0; i<n; i++)
      printf("%d \n", tat[i]);
printf("\n Wait time \n");
```

Process	BT(ms)	Priority	CT(ms)	TAT (ms)
P1	13	3	22	22
P2	5	2	9	9
P3	8	4	30	30
P4	4	1	4	4

Average Turnaround time :- 16.25ms

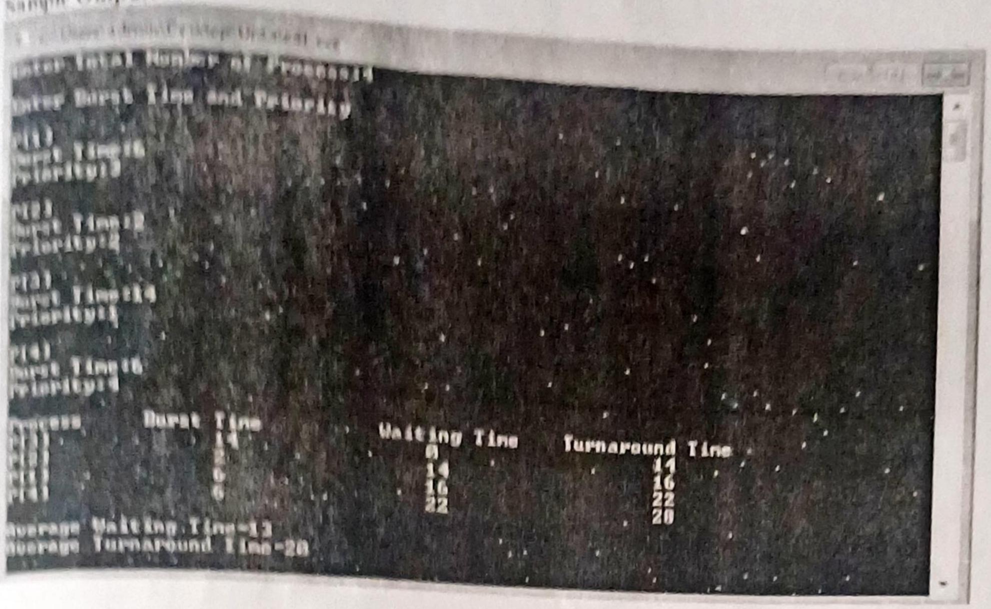
Average Wait time :- ~~8~~ 8.25ms

Gantt Chart :-

P4	P2	P1	P3

0 4 9 22 30

Sample Output:



```
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
P[3]               14                Waiting Time    Turnaround Time
P[2]               2                      0                   14
P[1]               6                      14                  16
P[4]               6                      16                  22
                                          22                  28
Average Waiting Time=13
Average Turnaround Time=20
```

Result:

The priority scheduling technique is implemented using C