Ex. No.: 6c)
Date:

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,temp;
    printf("Enter the number of process:");
    scanf("%d",&n);
    int p[3][n];
    for (int i=0;i<n;i++)
    {
        p[0][i]=i+1;
        printf("P[%d]\nBurst Time:",i+1);
        scanf("%d",&p[1][i]);
        printf("Priority:");
        scanf("%d",&p[2][i]);
    }


    for (int i=0;i<n-1;i++)
    {
        for (int j=0;j<n-i-1;j++)
        {
            if (p[2][j]>p[2][j+1])
            {
                for (int k=0;k<3;k++)
                {
                    temp=p[k][j];
                    p[k][j]=p[k][j+1];
                    p[k][j+1]=temp;
                }
            }
        }
    }

    int wt=0,tat=p[1][0];
    float avg_wt=0,avg_tat=0;
    printf("Process\t\tBurst Time\t\tWaiting Time\tTurnaround Time\n");
    for (int j=0;j<n;j++)
```

```
        {
            printf("P[%d]          %d          %d          %d\n",p[0][j],p[1][j],wt,tat);
            avg_wt+=wt;
            avg_tat+=tat;
            wt+=p[1][j];
            tat+=p[1][j+1];
        }
        printf("%.1f",avg_wt/n);
        printf("%.1f",avg_tat/n);

}
```

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          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
```

Output:

```
[student@localhost ~]$ cc priority.c
[student@localhost ~]$ ./a.out
Enter the number of process:4
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
13.020.0[student@localhost ~]$
```

Result:
Program is executed successfully and output is verified.