

4. Construct a scheduling program with C that selects the waiting process with the smallest execution time to execute next.

PROGRAM :

```
#include<stdio.h>

int main()
{
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;

    float avg_wt,avg_tat;

    printf("Enter number of process:");
    scanf("%d",&n);

    printf("\nEnter Burst Time:\n");

    for(i=0;i<n;i++)
    {
        printf("p%d:",i+1);
        scanf("%d",&bt[i]);

        p[i]=i+1;
    }

    for(i=0;i<n;i++)
    {
        pos=i;
        for(j=i+1;j<n;j++)
        {
            if(bt[j]<bt[pos])
                pos=j;
        }

        temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;

        temp=p[i];
        p[i]=p[pos];
```

```

        p[pos]=temp;
    }
    wt[0]=0;
    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];

        total+=wt[i];
    }
    avg_wt=(float)total/n;
    total=0;
    printf("\nProcesst   Burst Time   tWaitingTimeTurnaround Time");
    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\np%dtt  %dtt  %dtt%d",p[i],bt[i],wt[i],tat[i]);
    }
    avg_tat=(float)total/n;
    printf("\nnAverage Waiting Time=%f",avg_wt);
    printf("\nAverage Turnaround Time=%fn",avg_tat);
}

```

OUTPUT :

```

Enter number of process:3
nEnter Burst Time:np1:45
p2:32
p3:18
nProcesst   Burst Time   tWaitingTimeTurnaround Timenp3tt  18tt  0ttt18np2tt  32tt  18ttt50np1tt  45tt  50ttt95
nnAverage Waiting Time=22.666666nAverage Turnaround Time=54.333332n
-----
Process exited after 15.43 seconds with return value 0
Press any key to continue . . . |

```