Scheduling Algorithm Codes

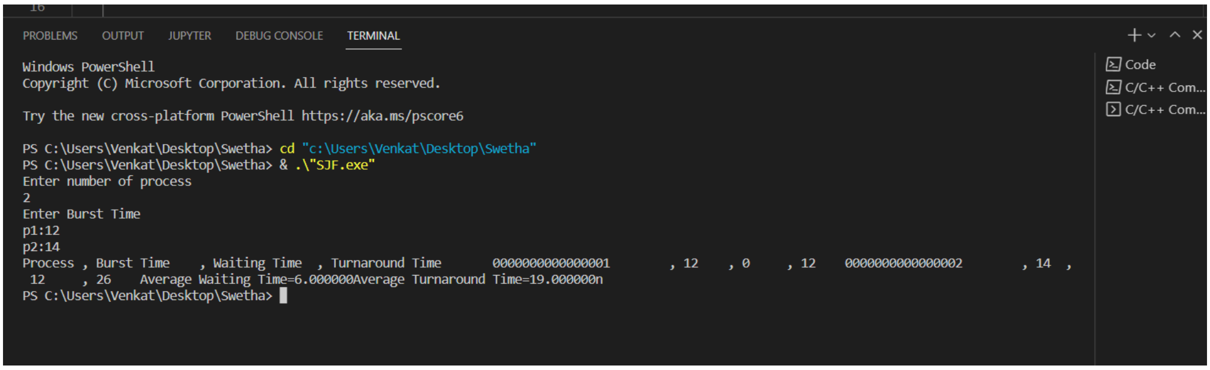
Swetha V

CB.EN.U4CYS21079

1. SJF

```
#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\n");
   scanf("%d",&n);

   printf("Enter 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("Process\t, Burst Time\t, Waiting Time\t, Turnaround Time\t");
for(i=0;i<n;i++)
{
    tat[i]=bt[i]+wt[i];
    total+=tat[i];
```

```
        printf("%p\t, %d\t, %d\t, %d\t",p[i],bt[i],wt[i],tat[i]);
    }

    avg_tat=(float)total/n;
    printf("Average Waiting Time=%f",avg_wt);
    printf("Average Turnaround Time=%fn",avg_tat);
}
```

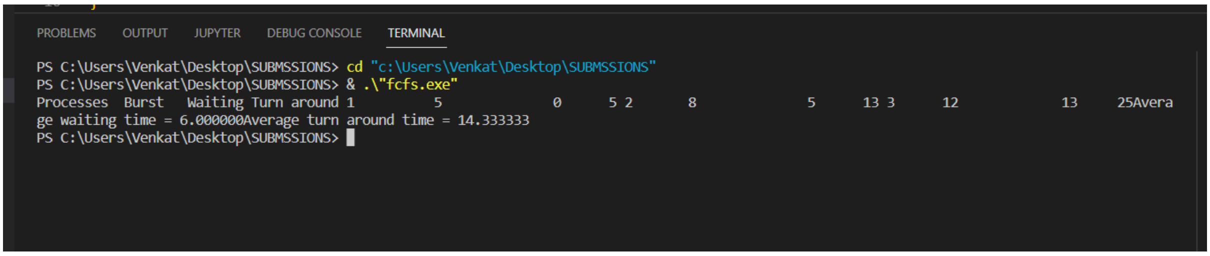


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Venkat\Desktop\Swetha> cd "c:\Users\Venkat\Desktop\Swetha"
PS C:\Users\Venkat\Desktop\Swetha> & .\"SJF.exe"
Enter number of process
2
Enter Burst Time
p1:12
p2:14
Process , Burst Time    , Waiting Time  , Turnaround Time       0000000000000001        , 12    , 0     , 12    0000000000000002        , 14  ,
 12     , 26    Average Waiting Time=6.000000Average Turnaround Time=19.000000n
PS C:\Users\Venkat\Desktop\Swetha> 
```

2. FCFS

```
#include <stdio.h>
int waitingtime(int proc[], int n,
int burst_time[], int wait_time[]) {
     wait_time[0] = 0;
      for (int i = 1; i < n ; i++ )
   wait_time[i] = burst_time[i-1] + wait_time[i-1] ;
   return 0;
}
int turnaroundtime( int proc[], int n,
int burst_time[], int wait_time[], int tat[]) {

   int i;
   for ( i = 0; i < n ; i++)
```

```
  tat[i] = burst_time[i] + wait_time[i];
  return 0;
}
int avgtime( int proc[], int n, int burst_time[]) {
  int wait_time[n], tat[n], total_wt = 0, total_tat = 0;
  int i;
    waitingtime(proc, n, burst_time, wait_time);
   turnaroundtime(proc, n, burst_time, wait_time, tat);
    printf("Processes  Burst   Waiting Turn around");
    for ( i=0; i<n; i++) {
    total_wt = total_wt + wait_time[i];
    total_tat = total_tat + tat[i];
    printf(" %d\t  %d\t\t %d \t%d", i+1, burst_time[i], wait_time[i], tat[i]);
  }
  printf("Average waiting time = %f", (float)total_wt / (float)n);
  printf("Average turn around time = %f", (float)total_tat / (float)n);
  return 0;
}
int main() {
    int proc[] = { 1, 2, 3};
  int n = sizeof proc / sizeof proc[0];
  //Burst time of all processes
  int burst_time[] = {5, 8, 12};
  avgtime(proc, n, burst_time);
  return 0;
}
```



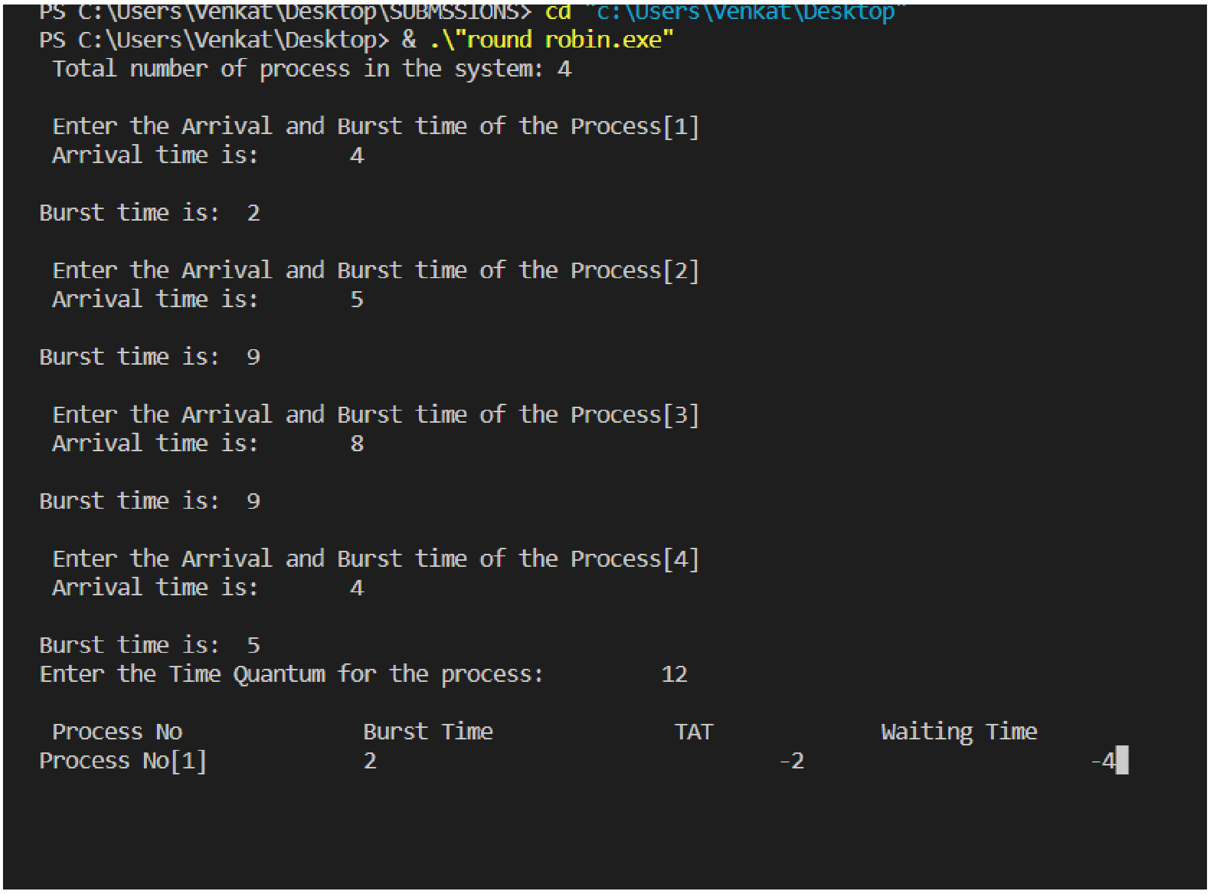
```
PS C:\Users\Venkat\Desktop\SUBMSSIONS> cd "c:\Users\Venkat\Desktop\SUBMSSIONS"
PS C:\Users\Venkat\Desktop\SUBMSSIONS> & .\"fcfs.exe"
Processes  Burst   Waiting Turn around 1         5             0      5 2       8              5      13 3      12             13      25Avera
ge waiting time = 6.000000Average turn around time = 14.333333
PS C:\Users\Venkat\Desktop\SUBMSSIONS> 
```

3. Round Robin

```
#include<stdio.h>

#include<conio.h>


void main()

{


   int i, NOP, sum=0,count=0, y, quant, wt=0, tat=0, at[10], bt[10], temp[10];

   float avg_wt, avg_tat;

   printf(" Total number of process in the system: ");

   scanf("%d", &NOP);

   y = NOP;



for(i=0; i<NOP; i++)

{

   printf("\n Enter the Arrival and Burst time of the Process[%d]\n", i+1);

printf(" Arrival time is: \t");

scanf("%d", &at[i]);

printf(" \nBurst time is: \t");

scanf("%d", &bt[i]);

temp[i] = bt[i];

}


printf("Enter the Time Quantum for the process: \t");
```

```
scanf("%d", &quant);
printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");
for(sum=0, i = 0; y!=0; )
{
if(temp[i] <= quant && temp[i] > 0)
{
   sum = sum + temp[i];
   temp[i] = 0;
   count=1;
   }
   else if(temp[i] > 0)
   {
      temp[i] = temp[i] - quant;
      sum = sum + quant;
   }
   if(temp[i]==0 && count==1)
   {
      y--;
      printf("\nProcess No[%d] \t\t %d\t\t\t\t %d\t\t\t %d", i+1, bt[i], sum-at[i], sum-at[i]-bt[i]);
      wt = wt+sum-at[i]-bt[i];
      tat = tat+sum-at[i];
      count =0;
   }
   if(i==NOP-1)
   {
      i=0;
   }
   else if(at[i+1]<=sum)
   {
      i++;
   }
```

```
    else
    {
        i=0;
    }
}

avg_wt = wt * 1.0/NOP;
avg_tat = tat * 1.0/NOP;
printf("\n Average Turn Around Time: \t%f", avg_wt);
printf("\n Average Waiting Time: \t%f", avg_tat);
getch();
}
```



```
PS C:\Users\Venkat\Desktop\SUBMSSIONS> cd "c:\Users\Venkat\Desktop"
PS C:\Users\Venkat\Desktop> & .\"round robin.exe"
 Total number of process in the system: 4

 Enter the Arrival and Burst time of the Process[1]
 Arrival time is:       4

Burst time is:  2

 Enter the Arrival and Burst time of the Process[2]
 Arrival time is:       5

Burst time is:  9

 Enter the Arrival and Burst time of the Process[3]
 Arrival time is:       8

Burst time is:  9

 Enter the Arrival and Burst time of the Process[4]
 Arrival time is:       4

Burst time is:  5
Enter the Time Quantum for the process:         12

 Process No             Burst Time              TAT             Waiting Time
Process No[1]           2                               -2                              -4
```