SCHOOL OF COMPUTER SCIENCE AND ENGINNERING

SUMBITTED TO LOVELY PROFESSION UNIVERSITY

In partial complete of the requirement of the award of

DEGREE OF BACHELOR OF TECHNOLOGY

{CSE}

NAME: K.SIDDARTHA

ROLL NO: 39

REGISTRATION NO: 12200271

SECTION : K22ZC

❖ <u>QUESTION</u>

Design a scheduler following non-preemptive scheduling approach to schedule the processes that arrives at different units and having burst time double the arrival time. Scheduler selects the process with largest burst time from the queue for the execution. Process is not being preempted until it finishes its service time. Compute the average waiting time and average turnaround time. What should be the average waiting time if processes are executed according to Shortest Job First scheduling approach with the same attribute values.

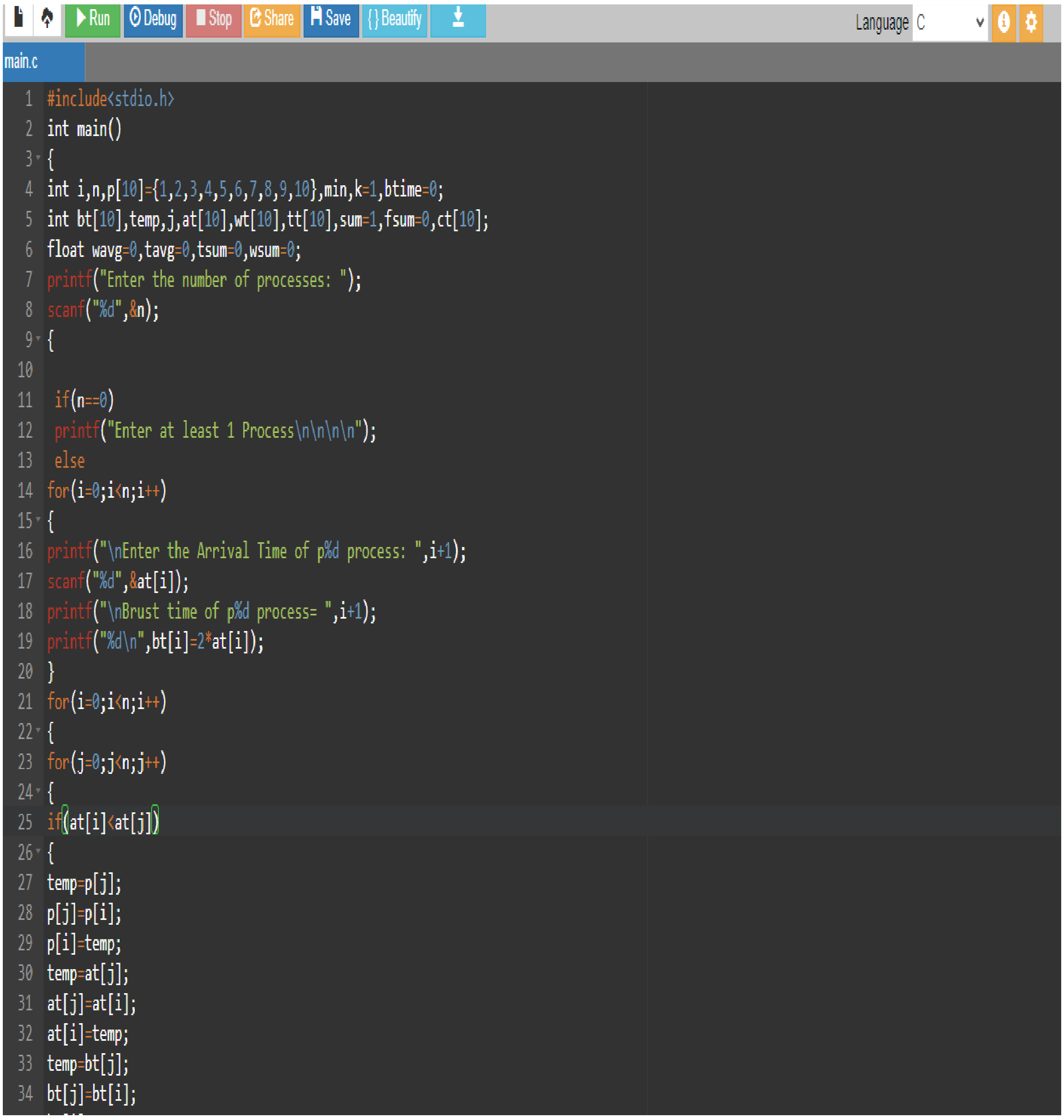
ANS:

```
#include<stdio.h>
int main()
{
int i,n,p[10]={1,2,3,4,5,6,7,8,9,10},min,k=1,btime=0;
int bt[10],temp,j,at[10],wt[10],tt[10],sum=1,fsum=0,ct[10];
float wavg=0,tavg=0,tsum=0,wsum=0;
printf("Enter the number of processes: ");
scanf("%d",&n);
{

 if(n==0)
 printf("Enter at least 1 Process\n\n\n\n");
 else
for(i=0;i<n;i++)
{
printf("\nEnter the Arrival Time of p%d process: ",i+1);
scanf("%d",&at[i]);
printf("\nBrust time of p%d process= ",i+1);
printf("%d\n",bt[i]=2*at[i]);
}
for(i=0;i<n;i++)                              //Sorting According to Arrival Time
{
for(j=0;j<n;j++)
{
if(at[i]<at[j])
```

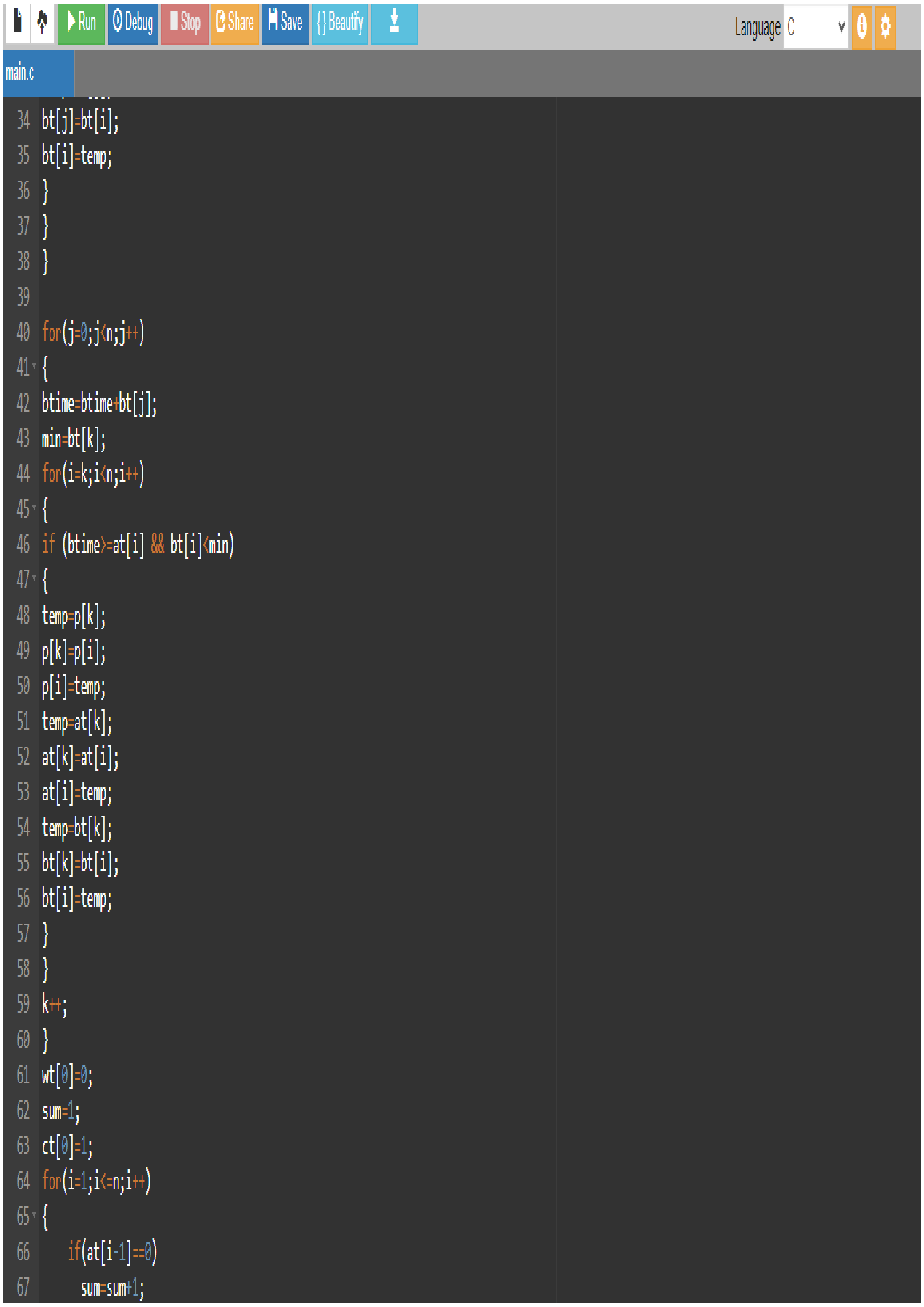
```
{
temp=p[j];
p[j]=p[i];
p[i]=temp;
temp=at[j];
at[j]=at[i];
at[i]=temp;
temp=bt[j];
bt[j]=bt[i];
bt[i]=temp;
}
}
}

for(j=0;j<n;j++)
{
btime=btime+bt[j];
min=bt[k];
for(i=k;i<n;i++)
{
if (btime>=at[i] && bt[i]<min)
{
temp=p[k];
p[k]=p[i];
p[i]=temp;
temp=at[k];
at[k]=at[i];

at[i]=temp;

temp=bt[k];

bt[k]=bt[i];

bt[i]=temp;

}

}

k++;

}

wt[0]=0;

sum=1;

ct[0]=1;

for(i=1;i<=n;i++)

{

            if(at[i-1]==0)

              sum=sum+1;

sum=sum+bt[i-1];

ct[i-1]=sum;

tt[i]=ct[i-1]-at[i-1];

tsum=tsum+tt[i];

wt[i]=tt[i]-bt[i-1];

wsum=wsum+wt[i];

}


wavg=(wsum/n);

tavg=(tsum/n);
```

```
printf("****************************************************");

printf("\n\t\t\t RESULT:-");

printf("\n\t   Shortest Job First,Non-Preemptive\n\n");

printf("\n\t\t-----------------------------------------------");

printf("\n\t\t|  Process  |\tBT\t|\tAT\t|\tWT\t|\tTAT\t|\tCT\t|" );

printf("\n\t\t-----------------------------------------------");

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

{

printf("\n\t\t|  p%d\t    |\t%d\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t|
",p[i],bt[i],at[i],wt[i+1],tt[i+1],ct[i]);

}

printf("\n\t\t-----------------------------------------------");



printf("\n\n\n***************************************************
*********");

 printf("\n\t\t\t\t Order of execution.\n");


printf("\n");


printf("\t\t\t\t\t--------------\n\t\t\t\t\t");


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

{

printf("| Process[%d] |\n\t\t\t\t\t",p[i]);

}
```

```c
printf("--------------\n\t\t\t\t\t");
printf("\n\n\n\nAverage Waiting Time: %f",wavg);
printf("\nAverage Turn Around Time: %f",tavg);
return 0;
}}
```

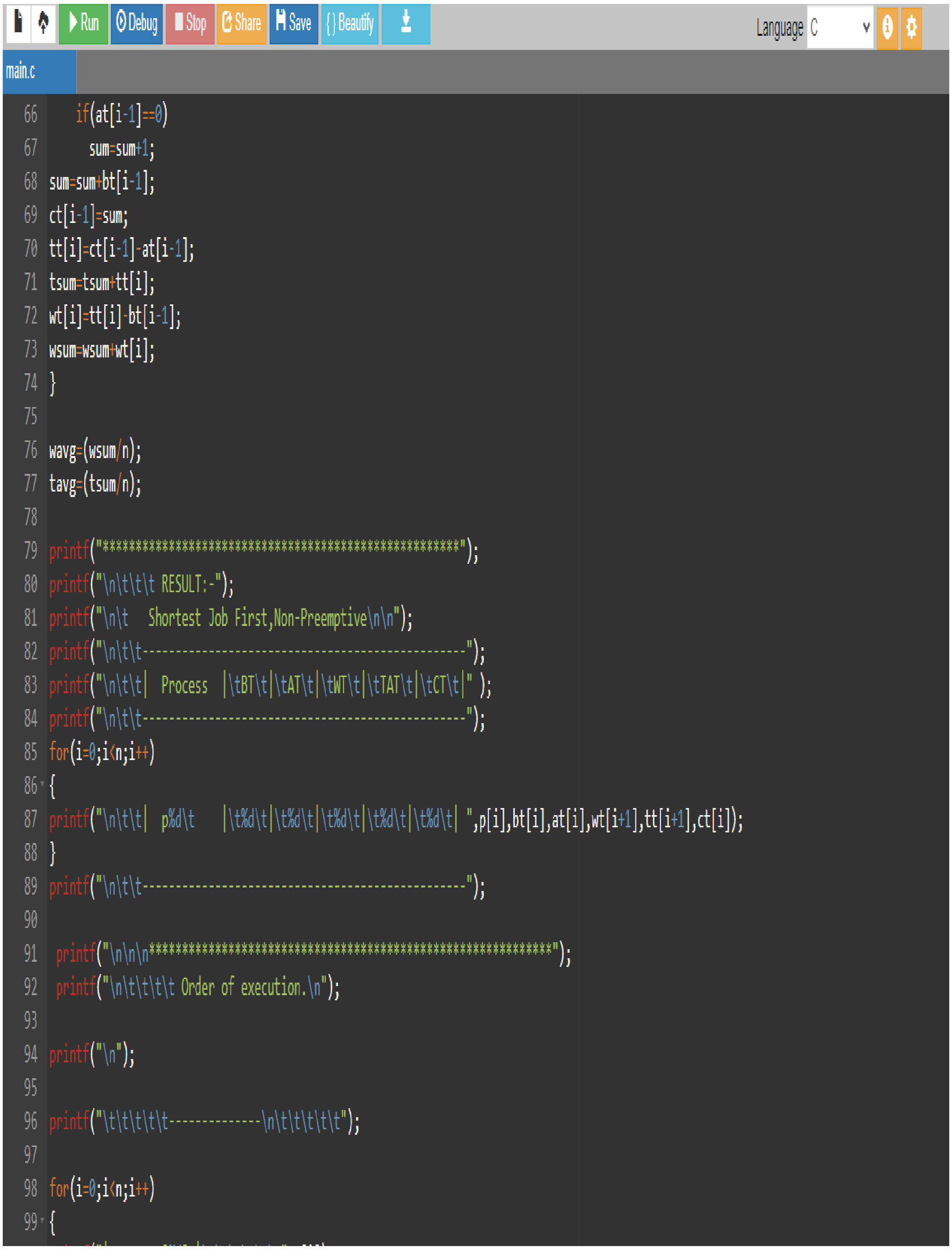


```c
#include<stdio.h>
int main()
{
int i,n,p[10]={1,2,3,4,5,6,7,8,9,10},min,k=1,btime=0;
int bt[10],temp,j,at[10],wt[10],tt[10],sum=1,fsum=0,ct[10];
float wavg=0,tavg=0,tsum=0,wsum=0;
printf("Enter the number of processes: ");
scanf("%d",&n);
{

 if(n==0)
 printf("Enter at least 1 Process\n\n\n\n");
 else
for(i=0;i<n;i++)
{
printf("\nEnter the Arrival Time of p%d process: ",i+1);
scanf("%d",&at[i]);
printf("\nBrust time of p%d process= ",i+1);
printf("%d\n",bt[i]=2*at[i]);
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
if(at[i]<at[j])
{
temp=p[j];
p[j]=p[i];
p[i]=temp;
temp=at[j];
at[j]=at[i];
at[i]=temp;
temp=bt[j];
bt[j]=bt[i];
```

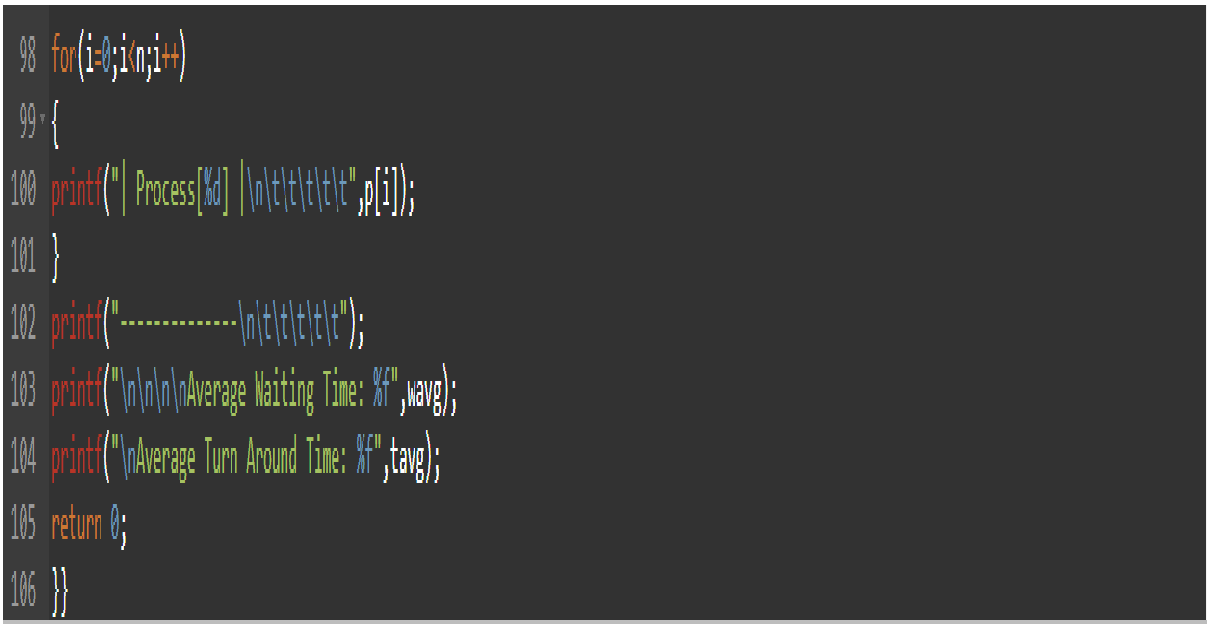


```
bt[j]=bt[i];
bt[i]=temp;
}
}
}

for(j=0;j<n;j++)
{
btime=btime+bt[j];
min=bt[k];
for(i=k;i<n;i++)
{
if (btime>=at[i] && bt[i]<min)
{
temp=p[k];
p[k]=p[i];
p[i]=temp;
temp=at[k];
at[k]=at[i];
at[i]=temp;
temp=bt[k];
bt[k]=bt[i];
bt[i]=temp;
}
}
k++;
}
wt[0]=0;
sum=1;
ct[0]=1;
for(i=1;i<=n;i++)
{
    if(at[i-1]==0)
      sum=sum+1;
```

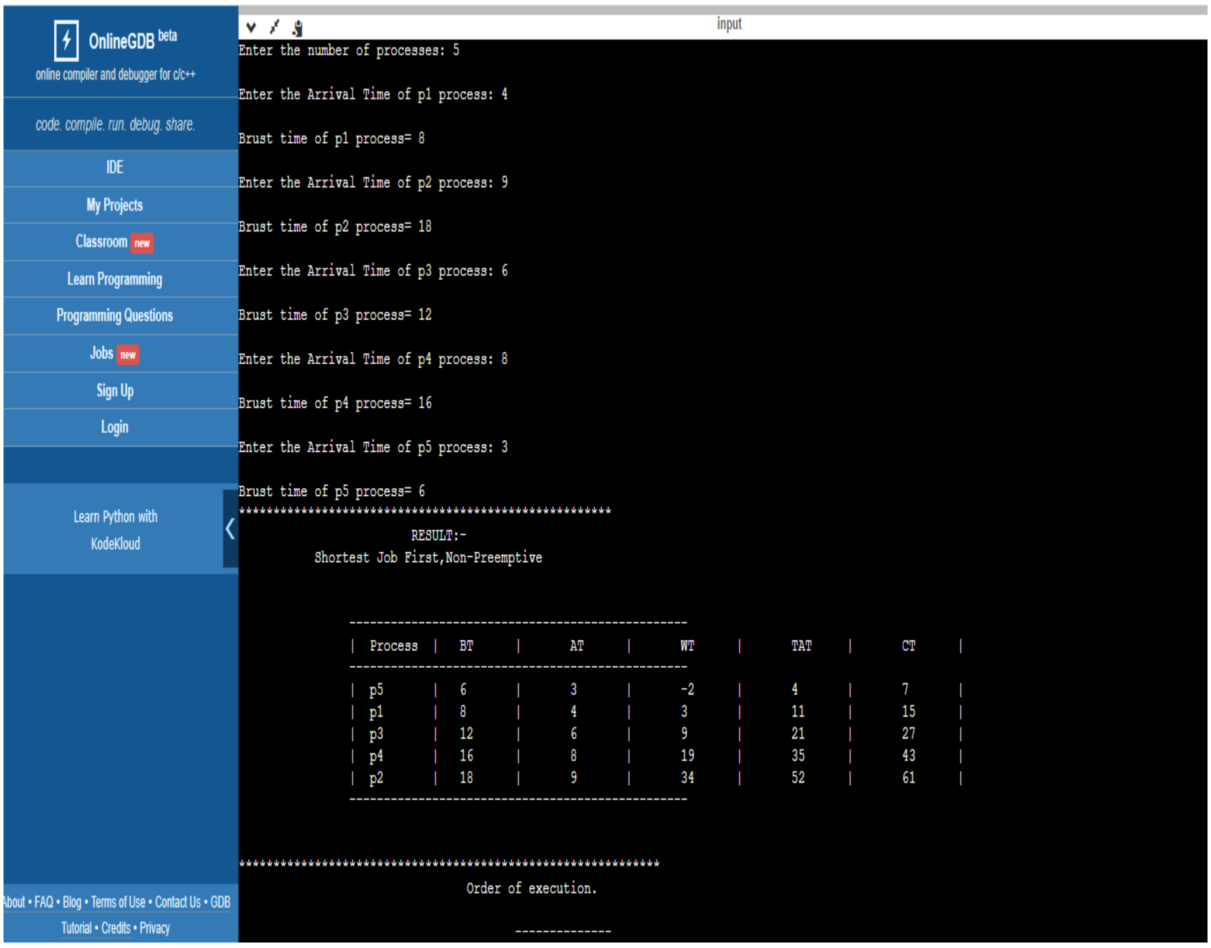
.



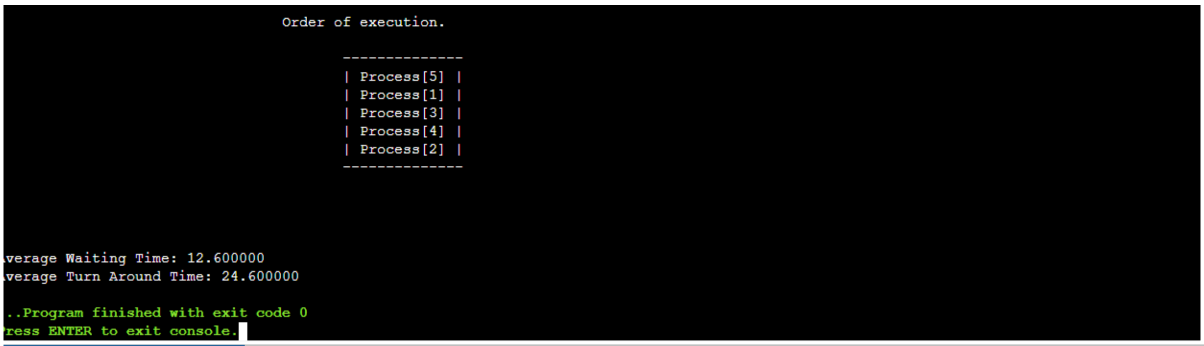
```c
    if(at[i-1]==0)
      sum=sum+1;
sum=sum+bt[i-1];
ct[i-1]=sum;
tt[i]=ct[i-1]-at[i-1];
tsum=tsum+tt[i];
wt[i]=tt[i]-bt[i-1];
wsum=wsum+wt[i];
}

wavg=(wsum/n);
tavg=(tsum/n);

printf("******************************************************");
printf("\n\t\t\t RESULT:-");
printf("\n\t   Shortest Job First,Non-Preemptive\n\n");
printf("\n\t\t------------------------------------------------");
printf("\n\t\t|  Process  |\tBT\t|\tAT\t|\tWT\t|\tTAT\t|\tCT\t|" );
printf("\n\t\t------------------------------------------------");
for(i=0;i<n;i++)
{
printf("\n\t\t|  p%d\t    |\t%d\t|\t%d\t|\t%d\t|\t%d\t|\t%d\t| ",p[i],bt[i],at[i],wt[i+1],tt[i+1],ct[i]);
}
printf("\n\t\t------------------------------------------------");

 printf("\n\n\n***************************************************************");
 printf("\n\t\t\t\t Order of execution.\n");

printf("\n");

printf("\t\t\t\t\t-------------\n\t\t\t\t\t");

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



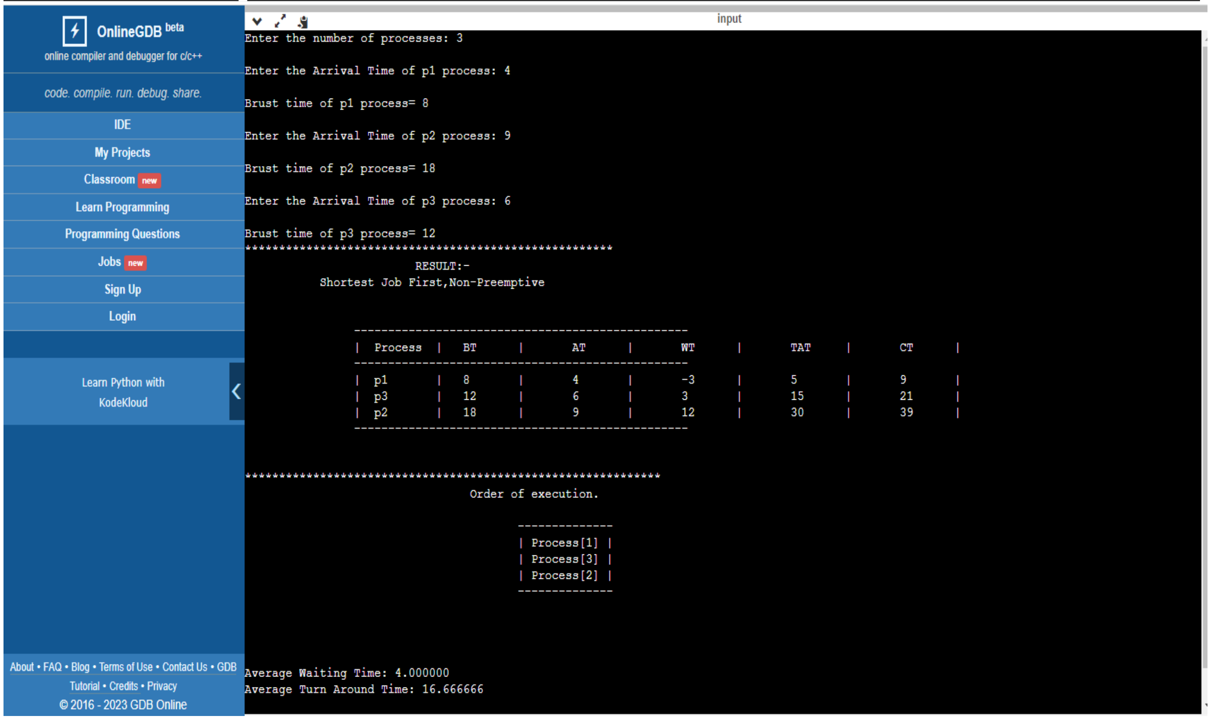
```c
for(i=0;i<n;i++)
{
printf("| Process[%d] |\n\t\t\t\t",p[i]);
}
printf("------------\n\t\t\t\t");
printf("\n\n\n\nAverage Waiting Time: %f",wavg);
printf("\nAverage Turn Around Time: %f",tavg);
return 0;
}}
```


OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
Learn Python with KodeKloud
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
input

```
Enter the number of processes: 5

Enter the Arrival Time of p1 process: 4

Brust time of p1 process= 8

Enter the Arrival Time of p2 process: 9

Brust time of p2 process= 18

Enter the Arrival Time of p3 process: 6

Brust time of p3 process= 12

Enter the Arrival Time of p4 process: 8

Brust time of p4 process= 16

Enter the Arrival Time of p5 process: 3

Brust time of p5 process= 6
*******************************************************
                         RESULT:-
           Shortest Job First,Non-Preemptive


                ------------------------------------------------
                |  Process  |  BT     |     AT     |     WT     |     TAT     |     CT     |
                ------------------------------------------------
                |  p5       |  6      |     3      |     -2     |     4       |     7      |
                |  p1       |  8      |     4      |     3      |     11      |     15     |
                |  p3       |  12     |     6      |     9      |     21      |     27     |
                |  p4       |  16     |     8      |     19     |     35      |     43     |
                |  p2       |  18     |     9      |     34     |     52      |     61     |
                ------------------------------------------------



**************************************************************
                                 Order of execution.

                                        -------------
```



```
                                Order of execution.

                                       ---------------
                                       | Process[5] |
                                       | Process[1] |
                                       | Process[3] |
                                       | Process[4] |
                                       | Process[2] |
                                       ---------------



verage Waiting Time: 12.600000
verage Turn Around Time: 24.600000

..Program finished with exit code 0
ress ENTER to exit console.
```


OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
Learn Python with KodeKloud
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2023 GDB Online
input

```
Enter the number of processes: 3

Enter the Arrival Time of p1 process: 4

Brust time of p1 process= 8

Enter the Arrival Time of p2 process: 9

Brust time of p2 process= 18

Enter the Arrival Time of p3 process: 6

Brust time of p3 process= 12
******************************************************
                         RESULT:-
           Shortest Job First,Non-Preemptive


                ------------------------------------------------
                |  Process  |   BT    |     AT     |      WT     |       TAT     |       CT     |
                ------------------------------------------------
                |  p1       |   8     |     4      |      -3     |       5       |       9      |
                |  p3       |   12    |     6      |      3      |       15      |       21     |
                |  p2       |   18    |     9      |      12     |       30      |       39     |
                ------------------------------------------------


*************************************************************
                                 Order of execution.

                                        --------------
                                        | Process[1] |
                                        | Process[3] |
                                        | Process[2] |
                                        --------------



Average Waiting Time: 4.000000
Average Turn Around Time: 16.666666
```