Write a C program to calculate average time and turnaround time of Priority and RoundRobin schedule

```c
#include <stdio.h>

void RoundRobin(int n, int p[], int bt[])
{
int i, sum = 0, count = 0, y, quant, wt = 0, tat = 0,
temp[10];
float avg_wt, avg_tat;
y = n;
for (i = 0; i < n; 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, sum - bt[i]);
wt = wt + sum - bt[i];
tat = tat + sum;
count = 0;
}
if (i == n - 1)
{
i = 0;
}
else
{
i++;
}
}
avg_wt = wt * 1.0 / n;
avg_tat = tat * 1.0 / n;
printf("\n Average Turn Around Time: \t%f", avg_wt);
printf("\n Average Waiting Time: \t%f", avg_tat);
}

void 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];
}
```

```
}
void turnaroundtime(int proc[], int n, int burst_time[],
int wait_time[], int tat[])
{
for (int i = 0; i < n; i++)
{
tat[i] = burst_time[i] + wait_time[i];
}
}
int avgtime(int proc[], int n, int burst_time[], int
prior[])
{
int wait_time[n], tat[n], total_wt = 0, total_tat = 0;

waitingtime(proc, n, burst_time, wait_time);
turnaroundtime(proc, n, burst_time, wait_time, tat);

printf("Process\tBurst Time\tpriority\tWaiting
Time\tTurnaround Time\n");

for (int i = 0; i < n; i++)
{
total_wt = total_wt + wait_time[i];
total_tat = total_tat + tat[i];
printf("%d\t\t%d\t%d\t\t%d\t\t%d\n", proc[i],
burst_time[i], prior[i], wait_time[i], tat[i]);
}

printf("Average waiting time = %f\n", (float)total_wt /
(float)n);
printf("Average turn around time = %f\n",
(float)total_tat / (float)n);
```

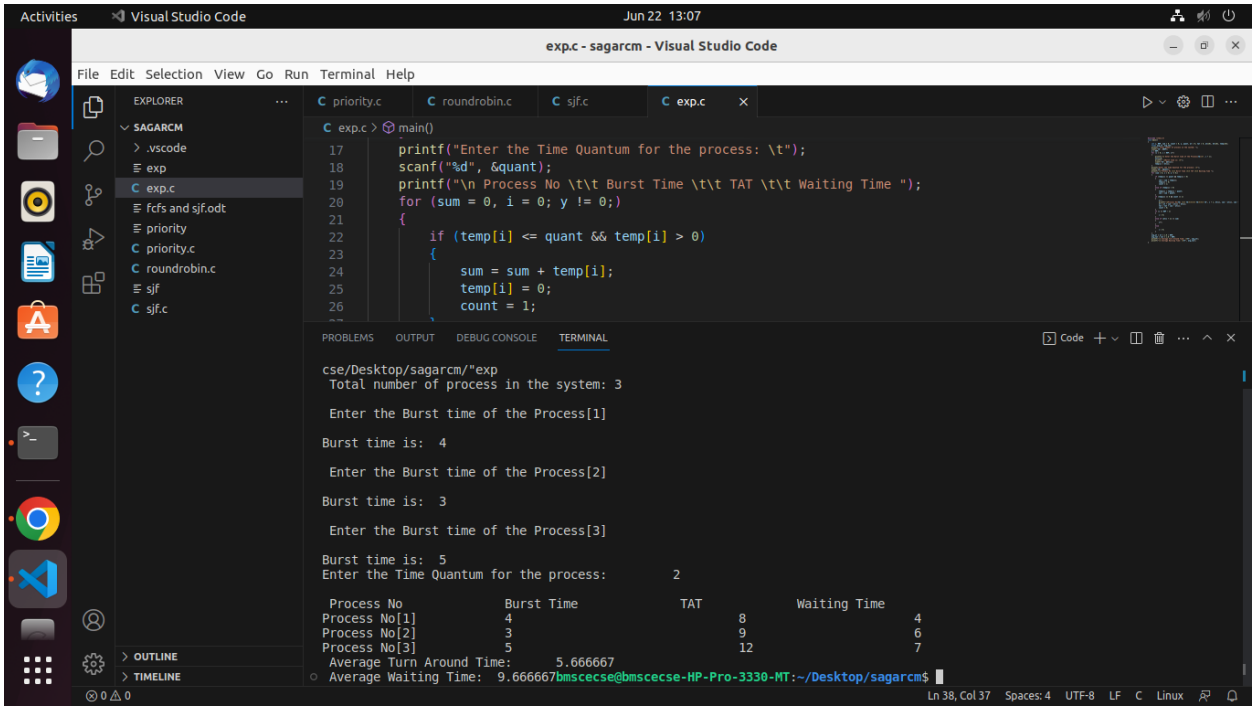
```
}
void main()
{
int ch = 0;
while (ch != 3)
{
printf("enter 1 for priority ,2 for RoundRobin\n");
scanf("%d", &ch);
switch (ch)
{
int j, temp2, temp, i, n, burst[10], proc[10], prior[10];
printf("Enter no of processes:");
scanf("%d", &n);
for (i = 0; i < n; i++)
{
printf("Enter burst time of process %d:", i + 1);
scanf("%d", &burst[i]);
proc[i] = i + 1;
}

case 1:
for (i = 0; i < n; i++)
{
printf("Enter priority time of process %d:", i + 1);
scanf("%d", &prior[i]);
}
for (i = 0; i < n; i++)
{
for (j = 0; j < n - i - 1; j++)
{
if (prior[j] > prior[j + 1])
{
```

```
temp = prior[j];
prior[j] = prior[j + 1];
prior[j + 1] = temp;
temp = burst[j];
burst[j] = burst[j + 1];
burst[j + 1] = temp;
temp2 = proc[j];
proc[j] = proc[j + 1];
proc[j + 1] = temp2;
}
}
}

avgtime(proc, n, burst, prior);
case 2:
RoundRobin(proc, burst, n);
case 3:
printf("program exits");
default :
printf("invalid choice");
}
}
}
```

Output


Activities
Visual Studio Code
Jun 22 13:07
exp.c - sagarcm - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
SAGARCM
.vscode
exp
exp.c
fcfs and sjf.odt
priority
priority.c
roundrobin.c
sjf
sjf.c
OUTLINE
TIMELINE
priority.c
roundrobin.c
sjf.c
exp.c
exp.c > main()
17 printf("Enter the Time Quantum for the process: \t");
18 scanf("%d", &quant);
19 printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");
20 for (sum = 0, i = 0; y != 0;)
21 {
22 if (temp[i] <= quant && temp[i] > 0)
23 {
24 sum = sum + temp[i];
25 temp[i] = 0;
26 count = 1;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
cse/Desktop/sagarcm/"exp
Total number of process in the system: 3
Enter the Burst time of the Process[1]
Burst time is: 4
Enter the Burst time of the Process[2]
Burst time is: 3
Enter the Burst time of the Process[3]
Burst time is: 5
Enter the Time Quantum for the process: 2
Process No Burst Time TAT Waiting Time
Process No[1] 4 8 4
Process No[2] 3 9 6
Process No[3] 5 12 7
Average Turn Around Time: 5.666667
Average Waiting Time: 9.666667bmscecse@bmscecse-HP-Pro-3330-MT:~/Desktop/sagarcm$
0 0
Ln 38, Col 37 Spaces: 4 UTF-8 LF C Linux