

8. Construct a C program to simulate Round Robin scheduling algorithm with C

A. Code:

```
#include<stdio.h>

#include<conio.h> int

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]);

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 %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();
}

```

Output:

```

Total number of process in the system: 4

Enter the Arrival and Burst time of the Process[1]
Arrival time is: 2
4

Enter the Arrival and Burst time of the Process[2]
Arrival time is: 3
5

Enter the Arrival and Burst time of the Process[3]
Arrival time is: 4
6

Enter the Arrival and Burst time of the Process[4]
Arrival time is: 5
7
Enter the Time Quantum for the process: 2

Process No      Burst Time      TAT      Waiting Time
Process No[1]   4                2         -2
Process No[2]   5                14         9
Process No[3]   6                15         9
Process No[4]   7                17        10
Average Turn Around Time: 6.500000
Average Waiting Time: 12.000000

```