

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

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  void main()
5  {
6      // initlialize the variable name
7      int i, NOP, sum=0, count=0, y, quant, wt=0, tat=0, at[10], bt[10], temp[10];
8      float avg_wt, avg_tat;
9      printf(" Total number of process in the system: ");
10     scanf("%d", &NOP);
11     y = NOP; // Assign the number of process to variable y
12
13     // Use for Loop to enter the details of the process Like Arrival time and the Burst Time
14     for(i=0; i<NOP; i++)
15     {
16         printf("\n Enter the Arrival and Burst time of the Process[%d]\n", i+1);
17         printf(" Arrival time is: \t"); // Accept arrival time
18         scanf("%d", &at[i]);
19         printf(" \nBurst time is: \t"); // Accept the Burst time
20         scanf("%d", &bt[i]);
21         temp[i] = bt[i]; // store the burst time in temp array
22     }
23     // Accept the Time qunat
24     printf("Enter the Time Quantum for the process: \t");
25     scanf("%d", &quant);
26     // Display the process No, burst time, Turn Around Time and the waiting time
27     printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");
28     for(sum=0, i = 0; y!=0; )
29     {
30         if(temp[i] <= quant && temp[i] > 0) // define the conditions
31         {
32             sum = sum + temp[i];
33             temp[i] = 0;
34             count=1;
35         }
36         else if(temp[i] > 0)
37         {
38             temp[i] = temp[i] - quant;
39             sum = sum + quant;
40         }
41         if(temp[i]==0 && count==1)
42             y--;
```

```

for(sum=0, i = 0; y!=0; )
{
    if(temp[i] <= quant && temp[i] > 0) // define the conditions
    {
        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--; //decrement the process no.
        printf("\nProcess No[%d] \t\t %d\t\t\t\t %d\t\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;
    }
}
// represents the average waiting time and Turn Around time
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:

```

C:\Users\kalya\OneDrive\Desktop\program\roundrobin.exe
Total number of process in the system: 3

Enter the Arrival and Burst time of the Process[1]
Arrival time is:      3

Burst time is: 3

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

Burst time is: 6

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

Burst time is: 9
Enter the Time Quantum for the process:      3

Process No      Burst Time      TAT      Waiting Time
Process No[1]    3      0      -3
Process No[2]    6      10     4
Process No[3]    9      16     7
Average Turn Around Time:      2.666667
Average Waiting Time: 8.666667

```