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

**PROGRAM:**

#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; // Assign the number of process to variable y

**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], sumat[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:**

