

**Ramakrishna Mission Vidyamandira**

(An Autonomous College under University of Calcutta)

Computer Science (Honors) Semester III 2022

Paper: CMSA CC 6 Practical

|  |
| --- |
| **Submitted By** |
| Class Roll Number: 302  Registration Number: A04-1112-0173-21  B.Sc.  3rd Semester  Batch: 2021-24 |

*// Write a program in C to demonstrate First Come First Serve (FCFS) scheduling algorithm and print the waiting times for each process and also print the average waiting time.*

*//Submitted By Roll : 302*

*//Source Code*

#include<stdio.h>

int main(){

    int bt[10]={0},at[10]={0},tat[10]={0},wt[10]={0},ct[10]={0};

    int n,sum=0;

    float totalTAT=0,totalWT=0;

    printf("Enter number of processes   ");

    scanf("%d",&n);

    printf("Enter arrival time and burst time for each process\n\n");

    for(int i=0;i<n;i++)

    {

        printf("Arrival time of process[%d] ",i+1);

        scanf("%d",&at[i]);

        printf("Burst time of process[%d]   ",i+1);

        scanf("%d",&bt[i]);

        printf("\n");

    }

*//calculate completion time of processes*

    for(int j=0;j<n;j++)

    {

        sum+=bt[j];

        ct[j]+=sum;

    }

*//calculate turnaround time and waiting times*

    for(int k=0;k<n;k++)

    {

        tat[k]=ct[k]-at[k];

        totalTAT+=tat[k];

    }

    for(int k=0;k<n;k++)

    {

        wt[k]=tat[k]-bt[k];

        totalWT+=wt[k];

    }

    printf("Solution: \n\n");

    printf("P#\t AT\t BT\t CT\t TAT\t WT\t\n\n");

    for(int i=0;i<n;i++)

    {

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

    }

    printf("\n\nAverage Turnaround Time = %f\n",totalTAT/n);

    printf("Average WT = %f\n\n",totalWT/n);

    return 0;

}

*////////////////////////OUTPUT//////////////////////*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS> gcc fcfs.c*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS> ./a.exe*

*// Enter number of processes:       5*

*// Enter arrival time and burst time for each process*

*// Arrival time of process[1]      3*

*// Burst time of process[1]        5*

*// Arrival time of process[2]      3*

*// Burst time of process[2]        6*

*// Arrival time of process[3]      4*

*// Burst time of process[3]        6*

*// Arrival time of process[4]      5*

*// Burst time of process[4]        7*

*// Arrival time of process[5]      4*

*// Burst time of process[5]        7*

*// Solution:*

*// P#       AT      BT      CT      TAT     WT*

*// P1       3       5       5       2       -3*

*// P2       3       6       11      8       2*

*// P3       4       6       17      13      7*

*// P4       5       7       24      19      12*

*// P5       4       7       31      27      20*

*// Average Turnaround Time = 13.800000*

*// Average WT = 7.600000*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS>*

*//  Write a program in C to demonstrate Shortest Job First (SJF) scheduling algorithm and print the waiting times for each process and also print the average waiting time.*

*//Submitted By Roll No: - 302*

*//Source Code*

#include <stdio.h>

int main()

{

    int bt[20], p[20], wt[20], tat[20], i, j, n, total = 0, pos, temp;

    float avg\_wt, avg\_tat;

    printf("Enter number of process:");

    scanf("%d", &n);

    printf("\nEnter Burst Time:\n");

    for (i = 0; i < n; i++)

    {

        printf("p%d:", i + 1);

        scanf("%d", &bt[i]);

        p[i] = i + 1;

    }

*//sorting of burst times*

    for (i = 0; i < n; i++)

    {

        pos = i;

        for (j = i + 1; j < n; j++)

        {

            if (bt[j] < bt[pos])

                pos = j;

        }

        temp = bt[i];

        bt[i] = bt[pos];

        bt[pos] = temp;

        temp = p[i];

        p[i] = p[pos];

        p[pos] = temp;

    }

    wt[0] = 0;

    for (i = 1; i < n; i++)

    {

        wt[i] = 0;

        for (j = 0; j < i; j++)

            wt[i] += bt[j];

        total += wt[i];

    }

    avg\_wt = (float)total / n;

    total = 0;

    printf("\nProcess\t    Burst Time    \tWaiting Time\tTurnaround Time");

    for (i = 0; i < n; i++)

    {

        tat[i] = bt[i] + wt[i];

        total += tat[i];

        printf("\np%d\t\t  %d\t\t    %d\t\t\t%d", p[i], bt[i], wt[i], tat[i]);

    }

    avg\_tat = (float)total / n;

    printf("\n\nAverage Waiting Time=%f", avg\_wt);

    printf("\nAverage Turnaround Time=%f\n", avg\_tat);

}

*/////////////////////OUTPUT////////////////*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS> ./a.exe*

*// Enter number of process:5*

*// Enter Burst Time:*

*// p1:5*

*// p2:6*

*// p3:7*

*// p4:4*

*// p5:3*

*// Process     Burst Time          Waiting Time    Turnaround Time*

*// p5                3                 0                   3*

*// p4                4                 3                   7*

*// p1                5                 7                   12*

*// p2                6                 12                  18*

*// p3                7                 18                  25*

*// Average Waiting Time=8.000000*

*// Average Turnaround Time=13.000000*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS>*

*// Write a program in C to demonstrate Round Robin (FCFS) scheduling algorithm and print the waiting times for each process and also print the average waiting time.*

*//Submitted By Roll No :- 302*

*//Source Code*

#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;

    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 %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);

    getchar();

}

*////////////////////OUTPUT//////////////*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS> gcc rr.c*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS> ./a.exe*

*//  Total number of process in the system: 5*

*//  Enter the Arrival and Burst time of the Process[1]*

*//  Arrival time is:       4*

*// Burst time is:  6*

*//  Enter the Arrival and Burst time of the Process[2]*

*//  Arrival time is:       3*

*// Burst time is:  5*

*//  Enter the Arrival and Burst time of the Process[3]*

*//  Arrival time is:       4*

*// Burst time is:  7*

*//  Enter the Arrival and Burst time of the Process[4]*

*//  Arrival time is:       2*

*// Burst time is:  5*

*//  Enter the Arrival and Burst time of the Process[5]*

*//  Arrival time is:       6*

*// Burst time is:  8*

*// Enter the Time Quantum for the process:         4*

*//  Process No              Burst Time              TAT             Waiting Time*

*// Process No[1]            6                       18                      12*

*// Process No[2]            5                       20                      15*

*// Process No[3]            7                       22                      15*

*// Process No[4]            5                       25                      20*

*// Process No[5]            8                       25                      17*

*//  Average Turn Around Time:      15.800000*

*//  Average Waiting Time:  22.000000*

*// PS C:\Users\Krishnendu Das\OneDrive\Desktop\Shell and OS>*