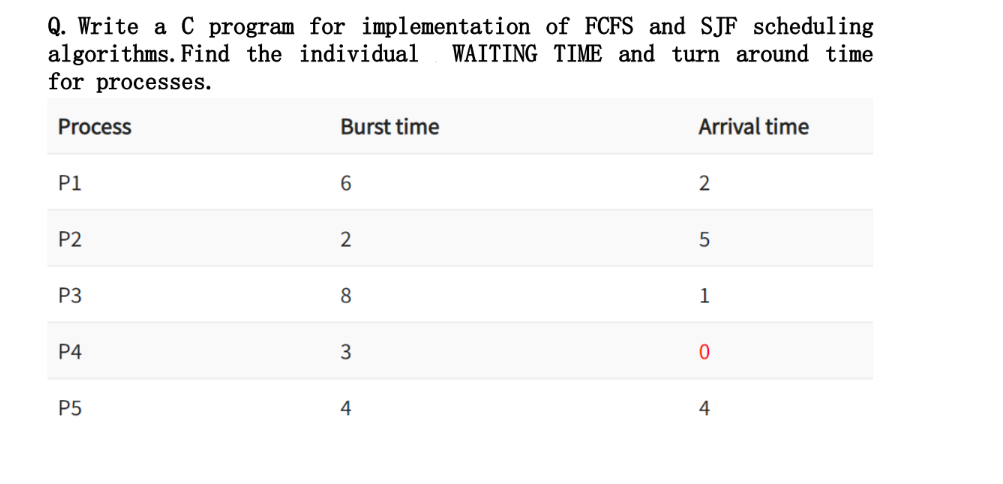
**NAME- AKSHAT JAISWAL**

**ROLL NO – 21052646**

**OS LAB (Scheduling Non-Preemptive)**

Q. Write a C program to implement FCFS and SJF scheduling algorithms. Find the individual WAITING TIME and turnaround time for processes.



**FCFS:**

#include <stdio.h>

*int* main(*void*)

{

*int* AT[5] = {2, 5, 1, 0, 4};

*int* BT[5] = {6, 2, 8, 3, 4};

*int* WT[5], CT[5], TAT[5];

*int* sum = 0;

*int* sumTAT = 0, sumWT = 0;

*int* min;

    printf("\nProcess\tAT\tBT\tCT\tTAT\tWT\n");

    //finding minimum arrival time

    for (*int* i = 0; i < 5; i++)

    {

        min = i;

        for (*int* j = i + 1; j < 5; j++)

        {

            if (AT[min] > AT[j])

            {

                min = j;

            }

        }

        //calculating

        sum += BT[min];

        CT[i] = sum;

        TAT[i] = CT[i] - AT[min];

        WT[i] = TAT[i] - BT[min];

        printf("%d\t%d\t%d\t%d\t%d\t%d\n", (i + 1), AT[min], BT[min], CT[i], TAT[i], WT[i]);

        sumTAT += TAT[i];

        sumWT += WT[i];

        //pushing the mimimum arrival time behind 'i' so it wont interfere

*int* temp = AT[min];

        AT[min] = AT[i];

        AT[i] = temp;

        //doing the same for burst time

        temp = BT[min];

        BT[min] = BT[i];

        BT[i] = temp;

    }

    printf("\nAverage Turnaround Time: %d\n", (sumTAT/5));

    printf("Average of Waiting Time: %d\n", (sumWT/5));

    return 0;

}

**Output:**

**Process AT BT CT TAT WT**

**1 0 3 3 3 0**

**2 1 8 11 10 2**

**3 2 6 17 15 9**

**4 4 4 21 17 13**

**5 5 2 23 18 16**

**Average Turnaround Time: 12**

**Average Waiting Time: 8**

**SJS:**

#include <stdio.h>

*void* swap(*int* \**arr*, *int* *i*, *int* *j*)

{

*int* temp = *arr*[*i*];

*arr*[*i*] = *arr*[*j*];

*arr*[*j*] = temp;

}

*int* main()

{

*int* BT[10], AT[10], P[10], WT[10], TAT[10], CT[10], sum = 0, n, totalWT = 0, totalTAT = 0, min, temp;

*float* avg\_wt, avg\_tat;

    printf("Enter number of process:");

    scanf("%d", &n);

    printf("\nEnter Burst Time:\n");

    for (*int* i = 0; i < n; i++)

    {

        printf("Process %d: ", i + 1);

        scanf("%d", &BT[i]);

        P[i] = i + 1;

    }

    printf("\nEnter Arrival Time:\n");

    for (*int* i = 0; i < n; i++)

    {

        printf("Process %d: ", i + 1);

        scanf("%d", &AT[i]);

    }

    // sorting according to arrival time

    for (*int* i = 0; i < n; i++)

    {

        for (*int* j = i + 1; j < n; j++)

        {

            if (AT[i] == AT[j])

            {

                if (BT[i] > BT[j])

                {

                    swap(BT, i, j);

                    swap(P, i, j);

                }

            }

            else if (AT[i] > AT[j])

            {

                swap(AT, i, j);

                swap(BT, i, j);

                swap(P, i, j);

            }

        }

    }

*int* run = 0;

    for (*int* i = 0; i < n; i++)

    {

        if (i == 0)

        {

            min = i;

        }

        else

        {

            // designing a code which gives the process to execute

            for (*int* j = i; j < run - 1; j++)

            {

                min = j;

                for (*int* k = j + 1; k < run; k++)

                {

                    if (k >= n)

                    {

                        goto label;

                    }

                    if (BT[min] > BT[k])

                    {

                        min = k;

                    }

                }

            }

        }

    label:

        sum += BT[min];

        CT[i] = sum;

        TAT[i] = CT[i] - AT[min];

        WT[i] = TAT[i] - BT[min];

        totalWT += WT[i];

        totalTAT += TAT[i];

        run += BT[min];

        // pushing the just executed process up so it doesn't not get repeated

        swap(AT, i, min);

        swap(BT, i, min);

        swap(P, i, min);

    }

    avg\_wt = (*float*)totalWT / n;

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

    for (*int* i = 0; i < n; 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*)totalTAT / n;

    printf("\n\nAverage Waiting Time=%.2f", avg\_wt);

    printf("\nAverage Turnaround Time=%.2f", avg\_tat);

}

**Enter number of process:5**

**Enter Burst Time:**

**Process 1: 6**

**Process 2: 2**

**Process 3: 8**

**Process 4: 3**

**Process 5: 4**

**Enter Arrival Time:**

**Process 1: 2**

**Process 2: 5**

**Process 3: 1**

**Process 4: 0**

**Process 5: 4**

**Process Burst Time Waiting Time Turnaround Time**

**p4 3 0 3**

**p1 6 1 7**

**p2 2 4 6**

**p5 4 7 11**

**p3 8 14 22**

**Average Waiting Time=5.20**

**Average Turnaround Time=9.80**