

1. Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time.

FCFS

SJF (pre-emptive & Non-pre-emptive)

Code:

```
#include <stdio.h>
int at[10], pt[10], ia, ip, n;
int tat[10], wt[10], it, iw, pos, j, i;
float atat = 0, awt = 0;
void fcfs()
{
    int t;
    printf("Enter number of processes: ");
    scanf("%d", &n);

    printf("Enter arrival times:\n");
    for (ia = 0; ia < n; ia++)
        scanf("%d", &at[ia]);

    printf("Enter process times:\n");
    for (ip = 0; ip < n; ip++)
        scanf("%d", &pt[ip]);

    if (at[0] == at[1])
    {
        t = pt[1];
        pt[1] = pt[0];
        pt[0] = t;
    }

    if (at[0] != 0)
        tat[0] = at[0];

    for (it = 0; it < n; it++)
        tat[it] = 0;

    int i = 0;
    for (it = 0; it < n; it++)
    {
        while (i <= it)
            tat[it] += pt[i++];
        i = 0;
    }
}
```

```

for (it = 0; it < n; it++)
    tat[it] = tat[it] - at[it];

for (ia = 0; ia < n; ia++)
    wt[ia] = tat[ia] - pt[ia];

for (i = 0; i < n; i++)
{
    atat += tat[i];
    awt += wt[i];
}

atat = atat / n;
awt = awt / n;

for (i = 0; i < n; i++)
{
    printf("P%d\t%d\t%d\n", i, tat[i], wt[i]);
}

printf("Average TAT=%.2f\nAverage WT=%.2f\n", atat, awt);
}

```

```

void srtf()
{
    int rt[10], endTime, i, smallest;
    int remain = 0, time, sum_wait = 0, sum_turnaround = 0;
    printf("Enter no of Processes : ");
    scanf("%d", &n);
    printf("Enter arrival times\n");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &at[i]);
    }
    printf("Enter Process times \n");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &pt[i]);
        rt[i] = pt[i];
    }
    rt[9] = 9999;
    for (time = 0; remain != n; time++)
    {
        smallest = 9;
        for (i = 0; i < n; i++)
        {

```

```

        if (at[i] <= time && rt[i] < rt[smallest] && rt[i] > 0)
        {
            smallest = i;
        }
    }
    rt[smallest]--;
    if (rt[smallest] == 0)
    {
        remain++;
        endTime = time + 1;
        printf("\nP%d %d %d", smallest + 1, endTime - at[smallest], endTime - pt[smallest] -
at[smallest]);
        sum_wait += endTime - pt[smallest] - at[smallest];
        sum_turnaround += endTime - at[smallest];
    }
}
printf("\n\nAverage waiting time = %f\n", sum_wait * 1.0 / n);
printf("Average Turnaround time = %f", sum_turnaround * 1.0 / n);
}

```

```

void sjf()
{
    int completed = 0;
    int currentTime = 0;
    int complete[n], ct[n];

    printf("Enter number of processes: ");
    scanf("%d", &n);

    printf("Enter arrival times:\n");
    for (int ia = 0; ia < n; ia++)
        scanf("%d", &at[ia]);

    printf("Enter process times:\n");
    for (int ip = 0; ip < n; ip++)
        scanf("%d", &pt[ip]);

    for (int i = 0; i < n; i++)
    {
        complete[i] = 0;
        ct[i] = 0;
    }

    while (completed != n)
    {
        int shortest = -1;

```

```

int min_bt = 9999;

for (int i = 0; i < n; i++)
{
    if (at[i] <= currentTime && complete[i] == 0)
    {
        if (pt[i] < min_bt)
        {
            min_bt = pt[i];
            shortest = i;
        }
        if (pt[i] == min_bt)
        {
            if (at[i] < at[shortest])
            {
                shortest = i;
            }
        }
    }
}

if (shortest == -1)
{
    currentTime++;
}
else
{
    ct[shortest] = currentTime + pt[shortest];
    tat[shortest] = ct[shortest] - at[shortest];
    wt[shortest] = tat[shortest] - pt[shortest];
    complete[shortest] = 1;
    completed++;
    currentTime = ct[shortest];
}

for (int i = 0; i < n; i++)
{
    atat += tat[i];
    awt += wt[i];
}

atat = atat / n;
awt = awt / n;

for (int i = 0; i < n; i++)

```

```

    {
        printf("P%d\t%d\t%d\n", i, tat[i], wt[i]);
    }

    printf("\nAverage TAT = %f\nAverage WT = %f\n", atat, awt);
}

void main()
{
    int op = 1, x;
    printf("1.FCFS \n2.SJF \n3.SRTF\n");
    scanf("%d", &x);
    switch (x)
    {
        case 1:
            fcfs();
            break;
        case 2:
            sjf();
            break;

        case 3:
            srtf();
            break;

        default:
            printf("Invalid option \n");
    }
}

```

Output:

```

PS D:\VS Code\OS> cd "d:\VS Code\OS" ; if ($?) { gcc os.c -o os } ; if ($?) { .\os }
1.FCFS
2.SJF
3.SRTF
1
Enter number of processes: 3
Enter arrival times:
0 0 1
Enter process times:
8 4 1
P0      4      0
P1      12     4
P2      12    11
Average TAT=9.33
Average WT=5.00

```

```

PS D:\VS Code\OS> cd "d:\VS Code\OS\" ; if ($?) { gcc os.c -o os } ; if ($?) { .\os }
1.FCFs
2.SJF
3.SRTF
2
Enter number of processes: 3
Enter arrival times:
0 0 1
Enter process times:
8 4 1
P0    13    5
P1     4     0
P2     4     3

Average TAT = 7.000000
Average WT = 2.666667
PS D:\VS Code\OS>

```

```

PS D:\VS Code\OS> cd "d:\VS Code\OS\" ; if ($?) { gcc os.c -o os } ; if ($?) { .\os }
1.FCFs
2.SJF
3.SRTF
3
Enter no of Processes : 3
Enter arrival times
0 0 1
Enter Process times
8 4 1

P3  1  0
P2  5  1
P1 13  5

Average waiting time = 2.000000
Average Turnaround time = 6.333333
PS D:\VS Code\OS>

```

```

void leftRec(int i) {
    if (i == 0) return;
    leftRec(i-1);
    cout << arr[i] << " ";
    return;
}

```

```

for (i = 0; i < n; i++) {
    if (arr[i] == arr[i+1] && arr[i] > arr[i+1]) {
        temp = arr[i];
        arr[i] = arr[i+1];
        arr[i+1] = temp;
        swap(arr[i], arr[i+1]);
        swap(arr[i], temp);
    }
}

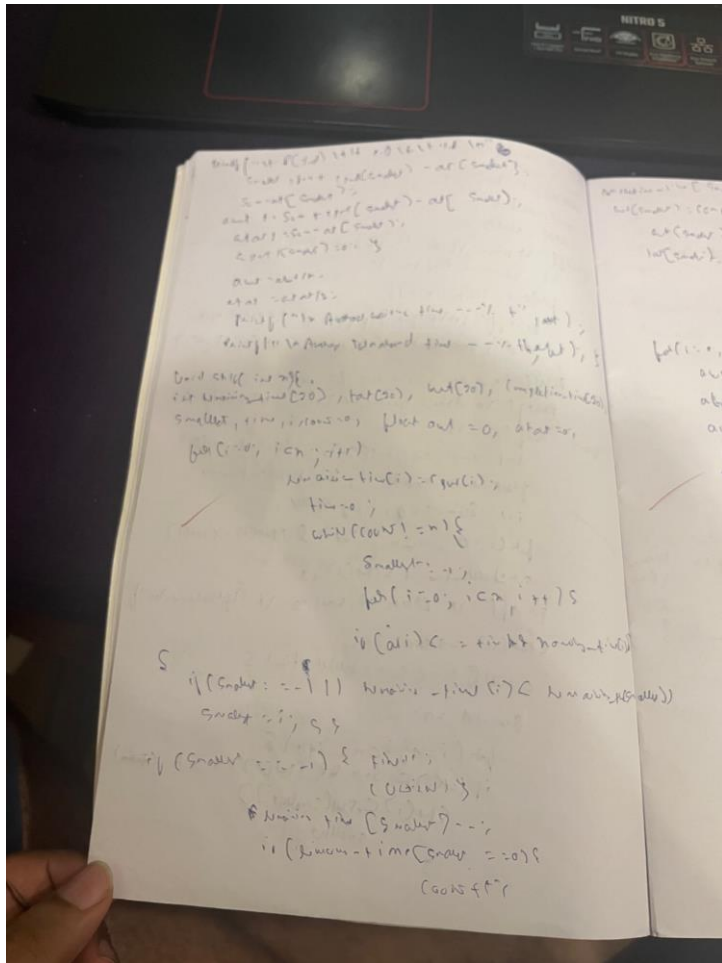
```

```

for (i = 0; i < n; i++) {
    sum = arr[i];
    arr[i] = sum;
    arr[i] = arr[i] - arr[i];
    arr[i] = arr[i] - arr[i];
}

```

[illegible]



Output

Enter the name of Phocoms

0 Run allowed time and CPU time for each

0 3
1 6
6 9
6 2

Ans

1) FCFS

2) ~~RR~~ SPT

3) SPT

4) ~~RR~~

Process	Arrival time	CPU time	Waiting time	Turnaround time
P0	0	3	0	3
P1	1	4	0	5
P2	4	6	0	10
P3	6	2	0	8

Avg waiting time -- 3.5000

Avg turnaround time -- 7.9500

2)

Phocoms

P0

P1

P2

P3

Avg waiting

Avg t

3)

Phocoms

0

1

2

3

Avg

Avg

1) FC

Output

Enter the name of processes

Enter arrival time and CPU time for each

0 3
1 6
2 5
3 2

Ans

1) FCFS

2) ~~SPT~~ SPT

3) SATF

4) SSTF

Process	Arrival time	CPU time	Waiting time
P0	0	3	0
P1	1	6	0
P2	4	5	0
P3	6	2	7

Avg waiting time -- 3.5000

Avg turnaround time -- 7.2500

2)

Process

P0

P1

P2

P3

P4

P5

P6

P7

P8

P9

P10

P11

P12

P13

P14

P15

P16

P17

P18

P19

P20

P21

P22

P23

P24

P25

P26

P27

P28

P29

P30

P31

P32

P33

P34

P35

P36

P37

P38

P39

P40

P41

P42

P43

P44

P45

P46

P47

P48

P49

P50

P51

P52

P53

P54

P55

P56

P57

P58

P59

P60

P61

P62

P63

P64

P65

P66

P67

P68

P69

P70

P71

P72

P73

P74

P75

P76

P77

P78

P79

P80

Process	Waiting time	Turnaround time
P(0)	0	0
P(1)	8	9
P(2)	9	17
P(3)	11	21

Avg waiting time = 6.7500

Avg turnaround time = 9.0000

Process	Arrival time	Service time	Waiting time	Turnaround time
0	0	3	0	3
1	1	6	0	14
2	4	4	2	10
3	6	2	2	8

Avg waiting time = 2.500

Avg turnaround time = 6.9500

~~FCFS~~

P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
0	1	2	3	4	5	6	7	8	9
P0(4)	P1(6)	P2(6)	P3(2)	P4(5)	P5(7)	P6(6)	P7(5)	P8(7)	P9(5)

71

p_0	p_0	p_1	p_1	p_1	p_2	p_2
0	1	2	3	4	5	6
$p(1)$	$p(2)$	$p(3)$	$p(4)$	$p(5)$	$p(6)$	15
	$p(1)$		$p(2)$	$p(3)$	$p(4)$	
		$p(1)$		$p(2)$		
			$p(1)$			

To 2000

1. *Stat*

Handed

inc

into

14

How

100

8

1

p_5	p_0	p_1	p_2	p_3	p_4
$p_5(1)$	$p_0(2)$	$p_1(3)$	$p_2(4)$	$p_3(5)$	$p_4(6)$
	$p_1(6)$	$p_2(7)$	$p_3(8)$	$p_4(9)$	$p_5(10)$

Write legibly

$$\frac{20}{20} \frac{20}{20} \frac{20}{20}$$

9/10

