Task 8: Write a program to find the finish time, turnaround time, and waiting time using FCFS Algorithm (input Takes by user).

Here's a basic example of a FCFS scheduling algorithm in C:

```
#include <stdio.h>


void findWaitingTime(int n, int bt[], int wt[], int at[]) {
    wt[0] = 0;
    for (int i = 1; i < n; i++) {
        wt[i] = bt[i - 1] + wt[i - 1] - at[i];
        if (wt[i] < 0) {
            wt[i] = 0;
        }
    }
}


void findTurnAroundTime(int n, int bt[], int wt[], int tat[]) {
    for (int i = 0; i < n; i++) {
        tat[i] = bt[i] + wt[i];
    }
}


void findFinishTime(int n, int at[], int bt[], int ft[]) {
    ft[0] = at[0] + bt[0];
    for (int i = 1; i < n; i++) {
        if (at[i] > ft[i - 1]) {
            ft[i] = at[i] + bt[i];
        } else {
            ft[i] = ft[i - 1] + bt[i];
        }
```

```
    }
}

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

    int burst_time[n], arrival_time[n], waiting_time[n], turnaround_time[n], finish_time[n];

    printf("Enter the burst time and arrival time for each process:\n");
    for (int i = 0; i < n; i++) {
        printf("Process %d: ", i + 1);
        scanf("%d %d", &burst_time[i], &arrival_time[i]);
    }

    findWaitingTime(n, burst_time, waiting_time, arrival_time);
    findTurnAroundTime(n, burst_time, waiting_time, turnaround_time);
    findFinishTime(n, arrival_time, burst_time, finish_time);

    printf("Process\tBurst Time\tArrival Time\tFinish Time\tTurnaround Time\tWaiting Time\n");
    for (int i = 0; i < n; i++) {
        printf("%d\t%d\t\t%d\t\t%d\t\t%d\t\t\t%d\n", i + 1, burst_time[i], arrival_time[i], finish_time[i],
turnaround_time[i], waiting_time[i]);
    }

    return 0;
}
//fcfs
```

Output-

```
Enter the number of processes: 3
Enter the burst time and arrival time for each process:
Process 1: 4 0
Process 2: 3 1
Process 3: 5 2
Process Burst Time      Arrival Time    Finish Time     Turnaround Time Waiting Time
1       4               0               4               4                       0
2       3               1               7               6                       3
3       5               2               12              9                       4
```