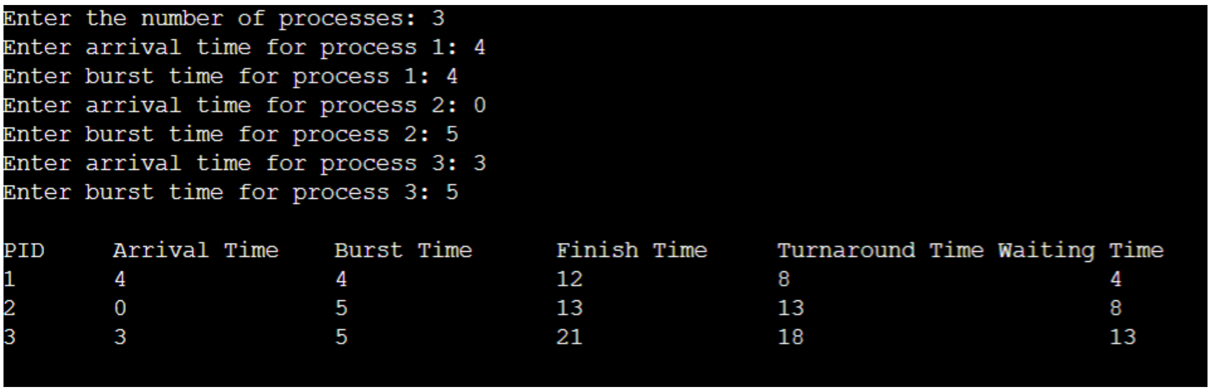
Task 9: Write a program to find the finish time, turnaround time, and waiting time using SJF Algorithm (input Takes by user).

```
 //sjf
#include <stdio.h>

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

    int complete = 0, t = 0, minm = 9999;
    int shortest = 0, finish_time;
    int flag = 0;

    while (complete != n) {
        for (int j = 0; j < n; j++) {
            if ((rt[j] <= t) && (rt[j] < minm) && (rt[j] > 0)) {
                minm = rt[j];
                shortest = j;
                flag = 1;
            }
        }

        if (flag == 0) {
            t++;
            continue;
        }

        rt[shortest]--;
        minm = rt[shortest];
        if (minm == 0) {
            minm = 9999;
        }

        if (rt[shortest] == 0) {
            complete++;
            flag = 0;
            finish_time = t + 1;
            wt[shortest] = finish_time - bt[shortest];
            if (wt[shortest] < 0) {
                wt[shortest] = 0;
            }
        }
        t++;
    }
}

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 wt[], int ft[]) {
    for (int i = 0; i < n; i++) {
        ft[i] = at[i] + bt[i] + wt[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];

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

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

    printf("\nPID\tArrival Time\tBurst 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, arrival_time[i], burst_time[i], finish_time[i],
turnaround_time[i], waiting_time[i]);
    }

    return 0;
}
```

Output-



```
Enter the number of processes: 3
Enter arrival time for process 1: 4
Enter burst time for process 1: 4
Enter arrival time for process 2: 0
Enter burst time for process 2: 5
Enter arrival time for process 3: 3
Enter burst time for process 3: 5

PID     Arrival Time    Burst Time      Finish Time     Turnaround Time Waiting Time
1       4               4               12              8                       4
2       0               5               13              13                      8
3       3               5               21              18                      13
```