Aim :

Design a CPU scheduling program with C using First Come First Served technique with the following considerations.

Algorithm :

1. Input Number of Processes: Prompt user for the number of processes.
2. Input Burst Times: For each process, prompt user for burst time and store it.
3. Calculate Waiting and Turnaround Times: Compute waiting time for each process based on previous processes' burst times. Calculate turnaround time for each process using its waiting time and burst time.
4. Print Process Details: Display process ID, burst time, waiting time, and turnaround time for each process.
5. Calculate and Print Average Times: Compute and print average waiting time and average turnaround time for all processes.
6. End Program: Exit the program.

Program & Output:

main.c

Run

```c
#include <stdio.h>

typedef struct {
    int id;
    int burstTime;
    int waitingTime;
    int turnaroundTime;
} Process;

void calculateTimes(Process proc[], int n) {
    proc[0].waitingTime = 0;
    for (int i = 1; i < n; i++) {
        proc[i].waitingTime = proc[i-1].waitingTime + proc[i-1].burstTime;
    }
    for (int i = 0; i < n; i++) {
        proc[i].turnaroundTime = proc[i].waitingTime + proc[i].burstTime;
    }
}

void printProcessDetails(Process proc[], int n) {
    int totalWaitingTime = 0, totalTurnaroundTime = 0;
    printf("PID\tBT\tWT\tTAT\n");
    for (int i = 0; i < n; i++) {
        totalWaitingTime += proc[i].waitingTime;
        totalTurnaroundTime += proc[i].turnaroundTime;
        printf("%d\t%d\t%d\t%d\n", proc[i].id, proc[i].burstTime, proc[i]
```

Output

```
/tmp/YXrGMJqbsU.o
Enter the number of processes: 5
Enter burst time for process 1: 123
Enter burst time for process 2: 345
3
Enter burst time for process 3: 34
Enter burst time for process 4: 1345
345
Enter burst time for process 5: PID BT  WT  TAT
1   1   0   1
2   3   1   4
3   34  4   38
4   345 38  383
5   345 383 728
Average Waiting Time = 85.20
Average Turnaround Time = 230.80


=== Code Execution Successful ===
```

Conclusion :

In conclusion, the program effectively calculates and displays the waiting time, turnaround time, and averages for a set of processes based on user-input burst times, demonstrating fundamental principles of process scheduling in operating systems.