LAB 2

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

- FCFS
- SJF

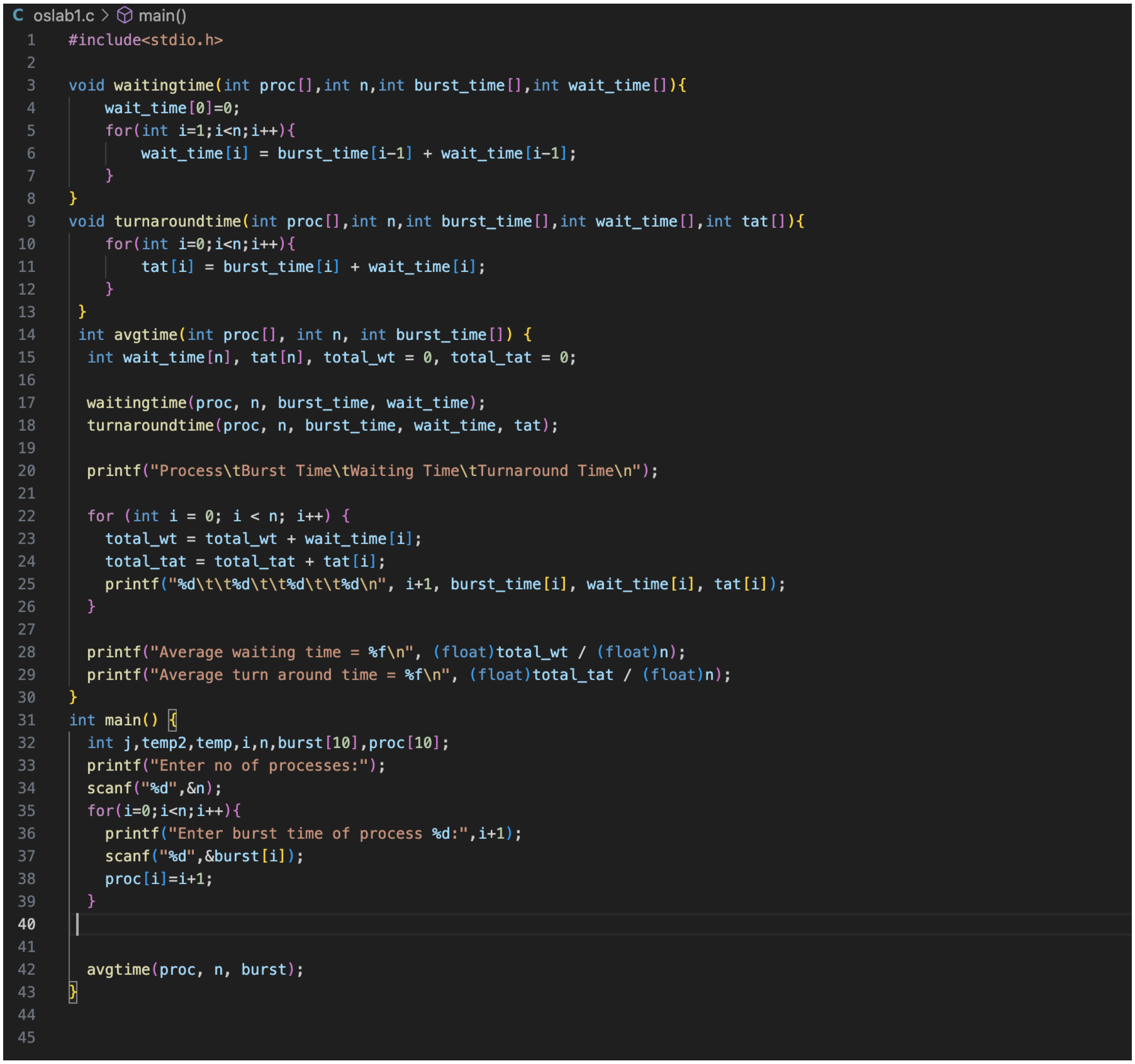
FCFS Scheduling:

Code:

```c
#include<stdio.h>

void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
    wait_time[0]=0;
    for(int i=1;i<n;i++){
        wait_time[i] = burst_time[i-1] + wait_time[i-1];
    }
}
void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[]){
    for(int i=0;i<n;i++){
        tat[i] = burst_time[i] + wait_time[i];
    }
 }
 int avgtime(int proc[], int n, int burst_time[]) {
  int wait_time[n], tat[n], total_wt = 0, total_tat = 0;

  waitingtime(proc, n, burst_time, wait_time);
  turnaroundtime(proc, n, burst_time, wait_time, tat);

  printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");

  for (int i = 0; i < n; i++) {
    total_wt = total_wt + wait_time[i];
    total_tat = total_tat + tat[i];
    printf("%d\t\t%d\t\t%d\t\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
  }

  printf("Average waiting time = %f\n", (float)total_wt / (float)n);
  printf("Average turn around time = %f\n", (float)total_tat / (float)n);
}
```

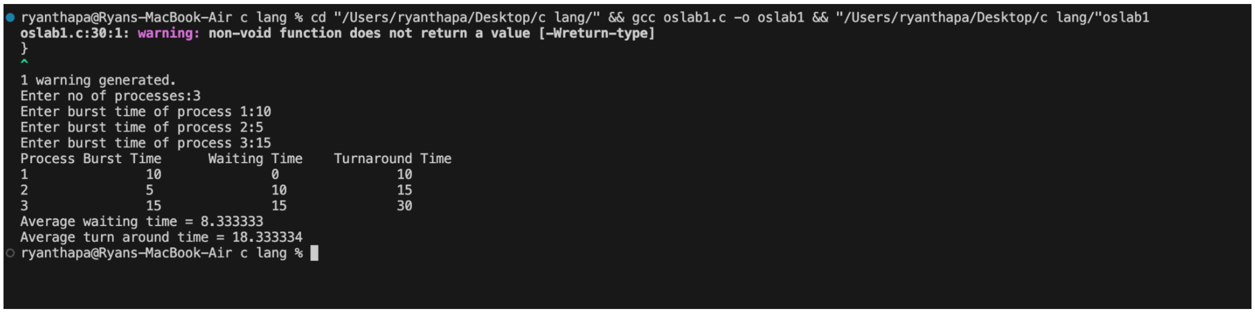
```
int main() {
  int j,temp2,temp,i,n,burst[10],proc[10];
  printf("Enter no of processes:");
  scanf("%d",&n);
  for(i=0;i<n;i++){
    printf("Enter burst time of process %d:",i+1);
    scanf("%d",&burst[i]);
    proc[i]=i+1;
  }
  avgtime(proc, n, burst);
}
```

Screenshots:



```
#include<stdio.h>

void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
    wait_time[0]=0;
    for(int i=1;i<n;i++){
        wait_time[i] = burst_time[i-1] + wait_time[i-1];
    }
}
void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[]){
    for(int i=0;i<n;i++){
        tat[i] = burst_time[i] + wait_time[i];
    }
 }
 int avgtime(int proc[], int n, int burst_time[]) {
  int wait_time[n], tat[n], total_wt = 0, total_tat = 0;

  waitingtime(proc, n, burst_time, wait_time);
  turnaroundtime(proc, n, burst_time, wait_time, tat);

  printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");

  for (int i = 0; i < n; i++) {
    total_wt = total_wt + wait_time[i];
    total_tat = total_tat + tat[i];
    printf("%d\t\t%d\t\t%d\t\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
  }

  printf("Average waiting time = %f\n", (float)total_wt / (float)n);
  printf("Average turn around time = %f\n", (float)total_tat / (float)n);
}
int main() {
  int j,temp2,temp,i,n,burst[10],proc[10];
  printf("Enter no of processes:");
  scanf("%d",&n);
  for(i=0;i<n;i++){
    printf("Enter burst time of process %d:",i+1);
    scanf("%d",&burst[i]);
    proc[i]=i+1;
  }


  avgtime(proc, n, burst);
}
```

Output:



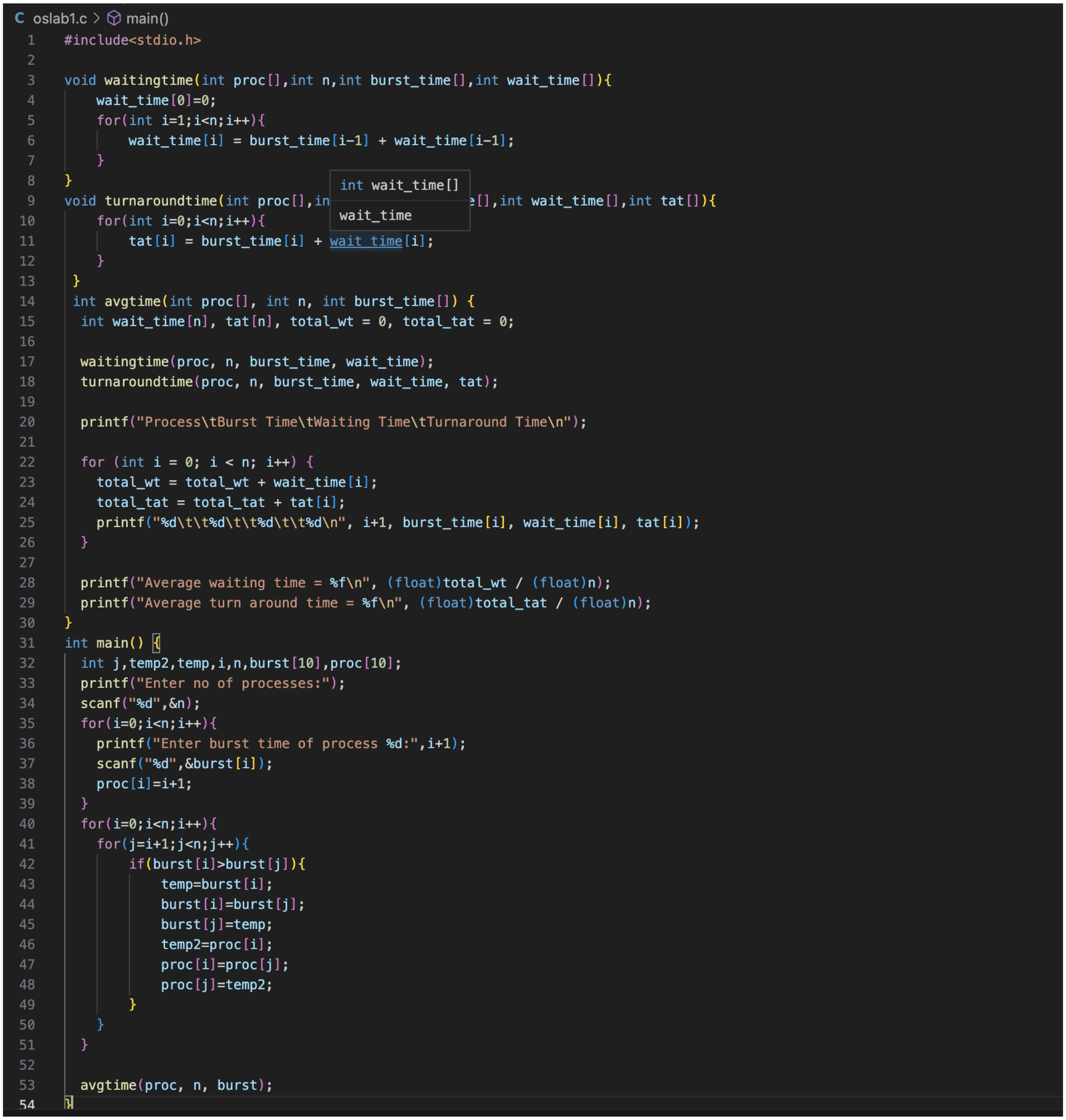
```
● ryanthapa@Ryans-MacBook-Air c lang % cd "/Users/ryanthapa/Desktop/c lang/" && gcc oslab1.c -o oslab1 && "/Users/ryanthapa/Desktop/c lang/"oslab1
  oslab1.c:30:1: warning: non-void function does not return a value [-Wreturn-type]
  }
  ^
  1 warning generated.
  Enter no of processes:3
  Enter burst time of process 1:10
  Enter burst time of process 2:5
  Enter burst time of process 3:15
  Process Burst Time      Waiting Time    Turnaround Time
  1               10              0               10
  2               5               10              15
  3               15              15              30
  Average waiting time = 8.333333
  Average turn around time = 18.333334
○ ryanthapa@Ryans-MacBook-Air c lang % 
```

SJF Scheduling:

Code:

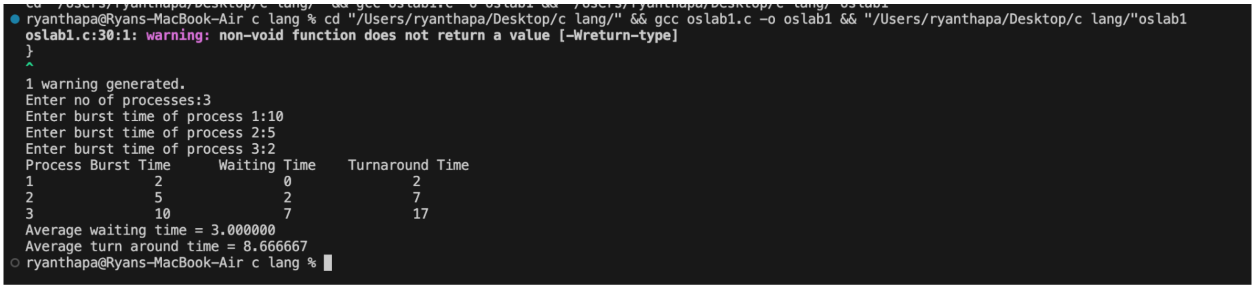
```
#include<stdio.h>

void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
    wait_time[0]=0;
    for(int i=1;i<n;i++){
        wait_time[i] = burst_time[i-1] + wait_time[i-1];
    }
}
void turnaroundtime(int proc[],int n,int burst_time[],int wait_time[],int tat[]){
    for(int i=0;i<n;i++){
        tat[i] = burst_time[i] + wait_time[i];
    }
 }
 int avgtime(int proc[], int n, int burst_time[]) {
  int wait_time[n], tat[n], total_wt = 0, total_tat = 0;

  waitingtime(proc, n, burst_time, wait_time);
  turnaroundtime(proc, n, burst_time, wait_time, tat);

  printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");

  for (int i = 0; i < n; i++) {
    total_wt = total_wt + wait_time[i];
    total_tat = total_tat + tat[i];
    printf("%d\t\t%d\t\t%d\t\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
  }

  printf("Average waiting time = %f\n", (float)total_wt / (float)n);
  printf("Average turn around time = %f\n", (float)total_tat / (float)n);
}
int main() {
  int j,temp2,temp,i,n,burst[10],proc[10];
  printf("Enter no of processes:");
  scanf("%d",&n);
  for(i=0;i<n;i++){
    printf("Enter burst time of process %d:",i+1);
    scanf("%d",&burst[i]);
    proc[i]=i+1;
  }
  for(i=0;i<n;i++){
    for(j=i+1;j<n;j++){
        if(burst[i]>burst[j]){
            temp=burst[i];
            burst[i]=burst[j];
            burst[j]=temp;
            temp2=proc[i];
            proc[i]=proc[j];
            proc[j]=temp2;
        }
    }
  }

  avgtime(proc, n, burst);
}
```

Screenshots:



```c
#include<stdio.h>

void waitingtime(int proc[],int n,int burst_time[],int wait_time[]){
    wait_time[0]=0;
    for(int i=1;i<n;i++){
        wait_time[i] = burst_time[i-1] + wait_time[i-1];
    }
}
void turnaroundtime(int proc[],in                  e[],int wait_time[],int tat[]){
    for(int i=0;i<n;i++){
        tat[i] = burst_time[i] + wait_time[i];
    }
 }
 int avgtime(int proc[], int n, int burst_time[]) {
  int wait_time[n], tat[n], total_wt = 0, total_tat = 0;

  waitingtime(proc, n, burst_time, wait_time);
  turnaroundtime(proc, n, burst_time, wait_time, tat);

  printf("Process\tBurst Time\tWaiting Time\tTurnaround Time\n");

  for (int i = 0; i < n; i++) {
    total_wt = total_wt + wait_time[i];
    total_tat = total_tat + tat[i];
    printf("%d\t\t%d\t\t%d\t\t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
  }

  printf("Average waiting time = %f\n", (float)total_wt / (float)n);
  printf("Average turn around time = %f\n", (float)total_tat / (float)n);
}
int main() {
  int j,temp2,temp,i,n,burst[10],proc[10];
  printf("Enter no of processes:");
  scanf("%d",&n);
  for(i=0;i<n;i++){
    printf("Enter burst time of process %d:",i+1);
    scanf("%d",&burst[i]);
    proc[i]=i+1;
  }
  for(i=0;i<n;i++){
    for(j=i+1;j<n;j++){
        if(burst[i]>burst[j]){
            temp=burst[i];
            burst[i]=burst[j];
            burst[j]=temp;
            temp2=proc[i];
            proc[i]=proc[j];
            proc[j]=temp2;
        }
    }
  }

  avgtime(proc, n, burst);
}
```

Output;



```
cd "/Users/ryanthapa/Desktop/c lang/" && gcc oslab1.c -o oslab1 && "/Users/ryanthapa/Desktop/c lang/"oslab1
● ryanthapa@Ryans-MacBook-Air c lang % cd "/Users/ryanthapa/Desktop/c lang/" && gcc oslab1.c -o oslab1 && "/Users/ryanthapa/Desktop/c lang/"oslab1
oslab1.c:30:1: warning: non-void function does not return a value [-Wreturn-type]
}
^

1 warning generated.
Enter no of processes:3
Enter burst time of process 1:10
Enter burst time of process 2:5
Enter burst time of process 3:2
Process Burst Time      Waiting Time    Turnaround Time
1               2               0               2
2               5               2               7
3               10              7               17
Average waiting time = 3.000000
Average turn around time = 8.666667
○ ryanthapa@Ryans-MacBook-Air c lang % █
```