1) FCFS Algorithm

```
#include<stdio.h>

int 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];
  }
}

int 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 proc[10];
```

```
  int n;
  int burst_time[10];

  printf("\nEnter the number of process: ");
  scanf("%d",&n);

  for(int i = 0 ; i < n ; i++){
    proc[i] = i+1;
    printf("\nEnter the Burst time for Process %d: ",i+1);
    scanf("%d",&burst_time[i]);
  }

  avgtime(proc, n, burst_time);
  return 0;
}
```

```
Enter the number of process: 3
Enter the Burst time for Process 1: 4
Enter the Burst time for Process 2: 3
Enter the Burst time for Process 3: 12
Process Burst Time  Waiting Time    Turnaround Time
1       4       0       4
2       3       4       7
3       12      7       19
Average waiting time = 3.666667
Average turn around time = 10.000000
```

2) SJF Algorithm

```
#include <stdio.h>

int main() {
  int bt[20], wt[20], tat[20], p[20], i, j, n, total = 0, pos, temp;
  float avg_wt, avg_tat;

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

  printf("Enter burst time for each process:\n");
  for (i = 0; i < n; i++) {
    printf("P[%d]: ", i + 1);
    scanf("%d", &bt[i]);
    p[i] = i + 1;
  }


  for (i = 0; i < n - 1; i++) {
    pos = i;
    for (j = i + 1; j < n; j++) {
      if (bt[j] < bt[pos]) {
        pos = j;
      }
    }
    temp = bt[i];
    bt[i] = bt[pos];
    bt[pos] = temp;

    temp = p[i];
    p[i] = p[pos];
    p[pos] = temp;
  }

  for (i = 0; i < n; i++) {
    wt[i] = 0;
```

```
    for (j = 0; j < i; j++) {
      wt[i] += bt[j];
    }
    total += wt[i];
  }

  avg_wt = (float)total / n;
  total = 0;

  printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");
  for (i = 0; i < n; i++) {
    tat[i] = bt[i] + wt[i];
    total += tat[i];
    printf("P[%d]\t\t%d\t\t%d\t\t%d\n", p[i], bt[i], wt[i], tat[i]);
  }

  avg_tat = (float)total / n;

  printf("\nAverage Waiting Time = %f", avg_wt);
  printf("\nAverage Turnaround Time = %f\n", avg_tat);

  return 0;
}
```

```
Enter number of processes: 3
4Enter burst time for each process:
P[1]: 4
P[2]: 3
P[3]: 12
Process Burst Time  Waiting Time    Turnaround Time
P[2]        3       0       3
P[1]        4       3       7
P[3]        12      7       19

Average Waiting Time = 3.333333
Average Turnaround Time = 9.666667
```