SJF Algorithm:

```c
#include<stdio.h>

int main() {
 int time, burst_time[10], at[10], sum_burst_time = 0, smallest, n, i;
 int sumt = 0, sumw = 0;
 printf("enter the no of processes : ");
 scanf("%d", & n);
 for (i = 0; i < n; i++) {
   printf("the arrival time for process P%d : ", i + 1);
   scanf("%d", & at[i]);
   printf("the burst time for process P%d : ", i + 1);
   scanf("%d", & burst_time[i]);
   sum_burst_time += burst_time[i];
 }
 burst_time[9] = 9999;
 for (time = 0; time < sum_burst_time;) {
   smallest = 9;
   for (i = 0; i < n; i++) {
     if (at[i] <= time && burst_time[i] > 0 && burst_time[i] <
burst_time[smallest])
       smallest = i;
   }
   printf("P[%d]\t|\t%d\t|\t%d\n", smallest + 1, time + burst_time[smallest] -
at[smallest], time - at[smallest]);
   sumt += time + burst_time[smallest] - at[smallest];
```
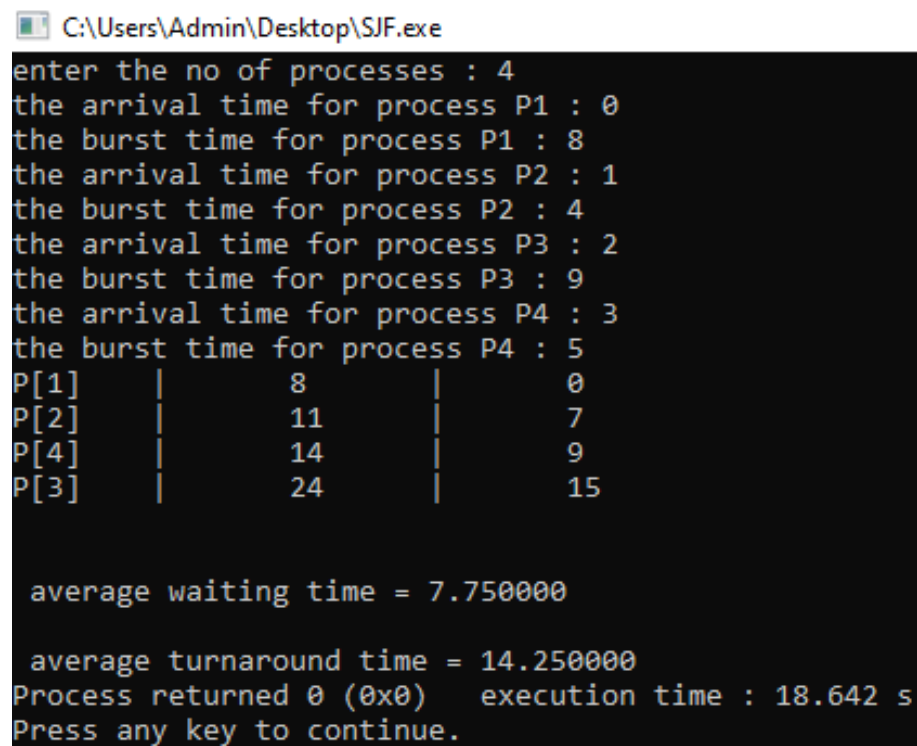
```
    sumw += time - at[smallest];

    time += burst_time[smallest];

    burst_time[smallest] = 0;

  }

  printf("\n\n average waiting time = %f", sumw * 1.0 / n);

  printf("\n\n average turnaround time = %f", sumt * 1.0 / n);

  return 0;

}
```

Output:

```
enter the no of processes : 4
the arrival time for process P1 : 0
the burst time for process P1 : 8
the arrival time for process P2 : 1
the burst time for process P2 : 4
the arrival time for process P3 : 2
the burst time for process P3 : 9
the arrival time for process P4 : 3
the burst time for process P4 : 5
P[1]    |       8        |       0
P[2]    |       11       |       7
P[4]    |       14       |       9
P[3]    |       24       |       15


 average waiting time = 7.750000

 average turnaround time = 14.250000
Process returned 0 (0x0)   execution time : 18.642 s
Press any key to continue.
```