## 4.SJF(SHORTEST JOB FIRST)

```
#include <stdio.h>
int main() {
    int n, i, j, temp;
    int bt[10], pid[10], wt[10], tat[10];
    float avg_wt = 0, avg_tat = 0;
 printf("Enter number of processes: ");
    scanf("%d", &n);
    for(i=0; i<n; i++) {
        pid[i] = i+1;
        printf("Burst time for P%d: ", pid[i]);
        scanf("%d", &bt[i]);
    }
    for(i=0; i<n-1; i++) {
        for(j=i+1; j<n; j++) {
            if(bt[i] > bt[j]) {
                temp = bt[i]; bt[i] = bt[j]; bt[j] = temp;
                temp = pid[i]; pid[i] = pid[j]; pid[j] = temp;
            }
        }
    }
    wt[0] = 0;
    for(i=1; i<n; i++)
        wt[i] = wt[i-1] + bt[i-1];
```

```
    for(i=0; i<n; i++)
        tat[i] = wt[i] + bt[i];
  for(i=0; i<n; i++) {
        avg_wt += wt[i];
        avg_tat += tat[i];
    }
    avg_wt /= n;
    avg_tat /= n;
    printf("\nP\tBT\tWT\tTAT\n");
    for(i=0; i<n; i++)
      printf("P%d\t%d\t%d\t%d\n", pid[i], bt[i], wt[i], tat[i]);
printf("\nAverage Waiting Time: %.2f", avg_wt);
    printf("\nAverage Turnaround Time: %.2f\n", avg_tat);
 return 0;
}
```