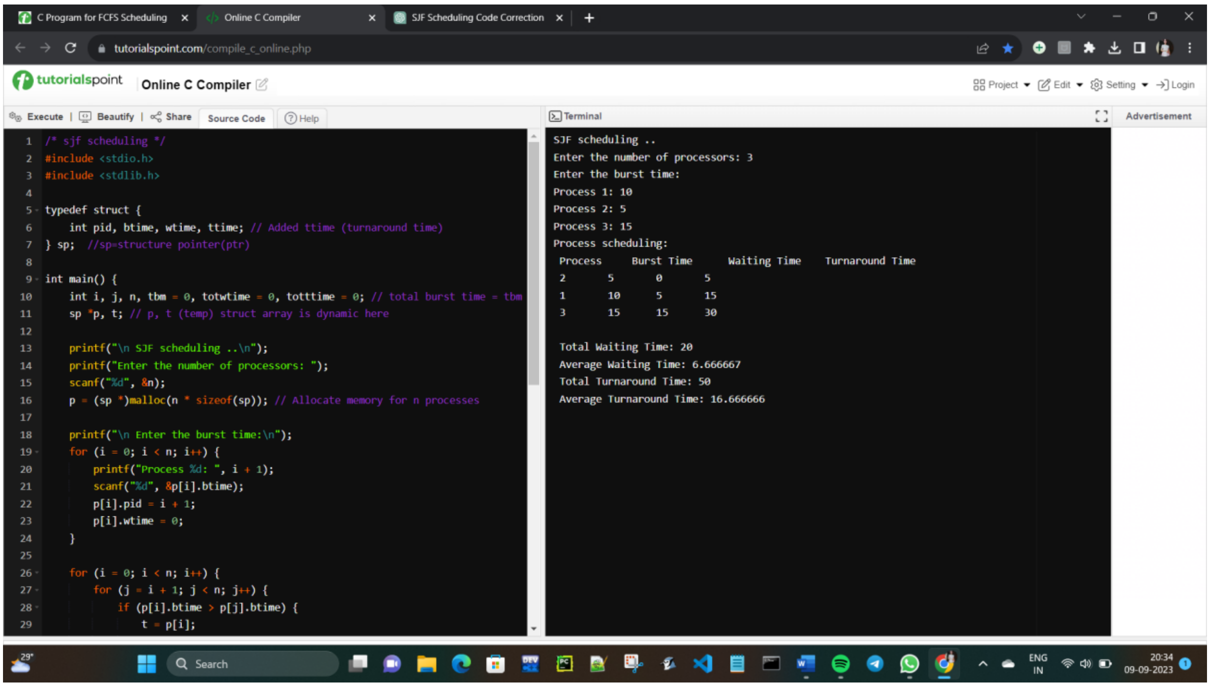
/* sjf scheduling (struct)(non preemptive)*/

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int pid, btime, wtime, ttime; // Added ttime (turnaround time)
} sp;  //sp=structure pointer(ptr)

int main() {
    int i, j, n, tbm = 0, totwtime = 0, totttime = 0; // total burst time = tbm
    sp *p, t; // p, t (temp) struct array is dynamic here

    printf("\n SJF scheduling ..\n");
    printf("Enter the number of processors: ");
    scanf("%d", &n);
    p = (sp *)malloc(n * sizeof(sp)); // Allocate memory for n processes

    printf("\n Enter the burst time:\n");
    for (i = 0; i < n; i++) {
        printf("Process %d: ", i + 1);
        scanf("%d", &p[i].btime);
        p[i].pid = i + 1;
        p[i].wtime = 0;
    }

    for (i = 0; i < n; i++) {
        for (j = i + 1; j < n; j++) {
            if (p[i].btime > p[j].btime) {
                t = p[i];
                p[i] = p[j];
                p[j] = t;
            }
        }
    }

    printf("\n Process scheduling:\n");
    printf(" Process \t Burst Time \t Waiting Time \t Turnaround Time\n");
    for (i = 0; i < n; i++) {
        tbm += p[i].btime;
        p[i].ttime = tbm;  //assigning total brust time to turn around time of p[i] as this whole is equal to tt of p[i]
        p[i].wtime = tbm - p[i].btime;  //as tt=bt+wt

        totwtime += p[i].wtime;
        totttime += p[i].ttime;

        printf(" %d\t\t %d\t\t %d\t\t %d\n", p[i].pid, p[i].btime, p[i].wtime, p[i].ttime);
    }

    printf("\n Total Waiting Time: %d\n", totwtime);
    printf(" Average Waiting Time: %f\n", (float)totwtime / n);
    printf(" Total Turnaround Time: %d\n", totttime);
    printf(" Average Turnaround Time: %f\n", (float)totttime / n);

    free(p); // Free allocated memory
    return 0;
}
```



```c
/* sjf scheduling */
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int pid, btime, wtime, ttime; // Added ttime (turnaround time)
} sp;  //sp=structure pointer(ptr)

int main() {
    int i, j, n, tbm = 0, totwtime = 0, totttime = 0; // total burst time = tbm
    sp *p, t; // p, t (temp) struct array is dynamic here

    printf("\n SJF scheduling ..\n");
    printf("Enter the number of processors: ");
    scanf("%d", &n);
    p = (sp *)malloc(n * sizeof(sp)); // Allocate memory for n processes

    printf("\n Enter the burst time:\n");
    for (i = 0; i < n; i++) {
        printf("Process %d: ", i + 1);
        scanf("%d", &p[i].btime);
        p[i].pid = i + 1;
        p[i].wtime = 0;
    }

    for (i = 0; i < n; i++) {
        for (j = i + 1; j < n; j++) {
            if (p[i].btime > p[j].btime) {
                t = p[i];
```

```
SJF scheduling ..
Enter the number of processors: 3
Enter the burst time:
Process 1: 10
Process 2: 5
Process 3: 15
Process scheduling:
 Process     Burst Time      Waiting Time    Turnaround Time
 2       5       0       5
 1       10      5       15
 3       15      15      30

 Total Waiting Time: 20
 Average Waiting Time: 6.666667
 Total Turnaround Time: 50
 Average Turnaround Time: 16.666666
```