

```
/* fcfs scheduling(struct) */
```

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

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

```
struct fcfs {
```

```
    int pid, btime, wtime, ttime;
```

```
};
```

```
int main() {
```

```
    int i, n;
```

```
    int totwtime = 0, totttime = 0;
```

```
    printf("\nFCFS scheduling..\n");
```

```
    printf("Enter the number of processes: ");
```

```
    scanf("%d", &n);
```

struct fcfs p[n]; //struct fcfs p[10]; declares an array named p that can hold n elements, and each element is of the struct fcfs type.

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

```
    //for the 0'th index of the structure array the process id(pid)=1(0+1)
```

```
    p[i].pid = i + 1; // Process IDs start from 1
```

```
    printf("\nBurst time of process %d: ", p[i].pid);
```

```
    scanf("%d", &p[i].btime);
```

```
}
```

```
// Calculate waiting time and turnaround time
```

```
for (i = 0; i < n; i++) { //index of struct array p is starting from 1
```

```
    if (i > 0) {
```

```
        p[i].wtime = p[i - 1].wtime + p[i - 1].btime;
```

```
    } else {
```

```

        p[i].wtime = 0; //for process residing at 1st id(pid=1)
    }

    p[i].ttime = p[i].wtime + p[i].btime; //for process residing at 1st id(pid=1)(tt=bt)
    totwtime += p[i].wtime;
    totttime += p[i].ttime;
}

printf("\nProcesses\tBurst\tWaiting\tTurnaround\n");
for (i = 0; i < n; i++) {
    printf("%d\t%d\t%d\t%d\n", p[i].pid, p[i].btime, p[i].wtime, p[i].ttime);
}

printf("\nAverage Waiting time = %.6f", (float)totwtime / n);
printf("\nAverage Turnaround time = %.6f\n", (float)totttime / n);

return 0;
}

```

The screenshot shows a web browser window with an online C compiler. The code is being executed, and the terminal output is visible on the right side of the editor.

Code Snippet:

```

22     printf("\nBurst time of process %d: ", p[i].pid);
23     scanf("%d", &p[i].btime);
24 }
25
26 // Calculate waiting time and turnaround time
27 for (i = 0; i < n; i++) { //index of struct array p is starting from 1
28     if (i > 0) {
29         p[i].wtime = p[i - 1].wtime + p[i - 1].btime;
30     } else {
31         p[i].wtime = 0; //for process residing at 1st id(pid=1)
32     }
33
34     p[i].ttime = p[i].wtime + p[i].btime; //for process residing at 1st id
35     (pid=1)(tt=bt)
36     totwtime += p[i].wtime;
37     totttime += p[i].ttime;
38 }
39
40 printf("\nProcesses\tBurst\tWaiting\tTurnaround\n");
41 for (i = 0; i < n; i++) {
42     printf("%d\t%d\t%d\t%d\n", p[i].pid, p[i].btime, p[i].wtime, p[i].ttime);
43 }
44
45 printf("\nAverage Waiting time = %.6f", (float)totwtime / n);
46 printf("\nAverage Turnaround time = %.6f\n", (float)totttime / n);
47
48 return 0;

```

Terminal Output:

```

FCFS scheduling..
Enter the number of processes: 3
Burst time of process 1: 5
Burst time of process 2: 10
Burst time of process 3: 15
Processes  Burst  Waiting  Turnaround
1          5      0         5
2          10     5         15
3          15     15         30

Average Waiting time = 6.666667
Average Turnaround time = 16.666666

```