

```
/* fcfs scheduling using at(array)*/
```

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

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

```
int main() {
```

```
    int i, n;
```

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

```
    printf("\n FCFS Scheduling\n");
```

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

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

```
    int pid[n], btime[n], wtime[n], ttime[n], atime[n];
```

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

```
        pid[i] = i + 1;
```

```
        printf("\nEnter burst time of process %d: ", pid[i]);
```

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

```
        printf("Enter arrival time of process %d: ", pid[i]);
```

```
        scanf("%d", &atime[i]);
```

```
    }
```

```
    // Sort the processes by their arrival times
```

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

```
        for (int j = i + 1; j < n; j++) {
```

```
            if (atime[i] > atime[j]) {
```

```
                int temp = atime[i];
```

```
                atime[i] = atime[j];
```

```
                atime[j] = temp;
```

```
                temp = btime[i];
```

```
                btime[i] = btime[j];
```

```

        btime[j] = temp;

        temp = pid[i];
        pid[i] = pid[j];
        pid[j] = temp;
    }
}

// Calculate waiting time and turnaround time
for (i = 0; i < n; i++) {
    if (i > 0) {
        wtime[i] = wtime[i - 1] + btime[i - 1] - atime[i];
        if (wtime[i] < 0) {
            wtime[i] = 0; // If the previous waiting time is negative, set it to 0 as waiting time can't be
negative
        }
    } else {
        wtime[i] = 0; // For the first process
    }

    ttime[i] = wtime[i] + btime[i];
    totwtime += wtime[i];
    totttime += ttime[i];
}

printf("\nProcesses\tBurst\tWaiting\tTurnaround\tArrival\n");

for (i = 0; i < n; i++) {
    printf("%d\t%d\t%d\t%d\t%d\n", pid[i], btime[i], wtime[i], ttime[i], atime[i]);
}

```

```
printf("\nAverage Waiting time = %f", (float)totwtime / n);
```

```
printf("\nAverage Turnaround time = %f\n", (float)totttime / n);
```

```
return 0;
```

```
}
```

The screenshot shows a web browser window with the URL `tutorialspoint.com/compile_c_online.php`. The page title is "Online C Compiler". The code editor contains the following C program for FCFS scheduling:

```
14 for (i = 0; i < n; i++) {
15     pid[i] = i + 1;
16     printf("\nEnter burst time of process %d: ", pid[i]);
17     scanf("%d", &btime[i]);
18     printf("\nEnter arrival time of process %d: ", pid[i]);
19     scanf("%d", &atime[i]);
20 }
21
22 // Sort the processes by their arrival times
23 for (i = 0; i < n; i++) {
24     for (int j = i + 1; j < n; j++) {
25         if (atime[i] > atime[j]) {
26             int temp = atime[i];
27             atime[i] = atime[j];
28             atime[j] = temp;
29
30             temp = btime[i];
31             btime[i] = btime[j];
32             btime[j] = temp;
33
34             temp = pid[i];
35             pid[i] = pid[j];
36             pid[j] = temp;
37         }
38     }
39 }
40
41 // Calculate waiting time and turnaround time
42 for (i = 0; i < n; i++) {
43     if (i > 0) {
```

The terminal output shows the execution of the program:

```
FCFS Scheduling
Enter the number of processes: 3
Enter burst time of process 1: 5
Enter arrival time of process 1: 2
Enter burst time of process 2: 10
Enter arrival time of process 2: 0
Enter burst time of process 3: 15
Enter arrival time of process 3: 1
Processes Burst Waiting Turnaround Arrival
2 10 0 10 0
3 15 9 24 1
1 5 22 27 2

Average Waiting time = 10.333333
Average Turnaround time = 20.333334
```