

main.c



Share

Run

Output

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 void fcfs(int requests[], int n, int start) {
4     int total_head_movement = 0, current_position = start;
5     for (int i = 0; i < n; i++) {
6         total_head_movement += abs(current_position - requests[i]);
7         current_position = requests[i];
8     }
9     printf("Total head movement: %d\n", total_head_movement);
10 }
11 int main() {
12     int requests[] = {176, 79, 34, 60, 92, 11, 41, 114};
13     int n = sizeof(requests) / sizeof(requests[0]);
14     int start = 50;
15
16     fcfs(requests, n, start);
17
18     return 0;
19 }
20
```

Total head movement: 510

=== Code Execution Successful ===

```
main.c
1 #include <stdio.h>
2 void findWaitingTime(int processes[], int n, int bt[], int wt[]) {
3     wt[0] = 0;
4     for (int i = 1; i < n; i++) {
5         wt[i] = bt[i - 1] + wt[i - 1];
6     }
7 }
8 void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) {
9     for (int i = 0; i < n; i++) {
10         tat[i] = bt[i] + wt[i];
11     }
12 }
13 void findavgTime(int processes[], int n, int bt[]) {
14     int wt[n], tat[n];
15     findWaitingTime(processes, n, bt, wt);
16     findTurnAroundTime(processes, n, bt, wt, tat);
17     printf("Process\tBurst Time\tWaiting Time\tTurn-Around Time\n");
18     int total_wt = 0, total_tat = 0;
19     for (int i = 0; i < n; i++) {
20         printf("%d\t\t%d\t\t%d\t\t%d\n", processes[i], bt[i], wt[i], tat[i]);
21         total_wt += wt[i];
22         total_tat += tat[i];
23     }
24     printf("\nAverage Waiting Time: %.2f\n", (float)total_wt / n);
25     printf("Average Turnaround Time: %.2f\n", (float)total_tat / n);
}
```

Output

Process	Burst Time	Waiting Time	Turn-Around Time
1	10	0	10
2	5	10	15
3	8	15	23

Average Waiting Time: 8.33  
Average Turnaround Time: 16.00

=== Code Execution Successful ===