

```

9  #include<stdio.h>
10 struct priority_scheduling {
11     char process_name;
12     int burst_time;
13     int waiting_time;
14     int turn_around_time;
15     int priority;
16 };
17 int main() {
18     int number_of_process;
19     int total = 0;
20     struct priority_scheduling temp_process;
21     int ASCII_number = 65;
22     int position;
23     float average_waiting_time;
24     float average_turnaround_time;
25     printf("Enter the total number of Processes: ");
26     scanf("%d", & number_of_process);
27     struct priority_scheduling process[number_of_process];
28     printf("\nPlease Enter the Burst Time and Priority of each process:\n");
29     for (int i = 0; i < number_of_process; i++) {
30         process[i].process_name = (char) ASCII_number;
31         printf("\nEnter the details of the process %c \n", process[i].process_name);
32         printf("Enter the burst time: ");
33         scanf("%d", & process[i].burst_time);
34         printf("Enter the priority: ");
35         scanf("%d", & process[i].priority);
36         ASCII_number++;
37     }
38     for (int i = 0; i < number_of_process; i++) {
39         position = i;
40         for (int j = i + 1; j < number_of_process; j++){
41             if (process[j].priority > process[position].priority)

```



```

36     ASCII_number++;
37 }
38 for (int i = 0; i < number_of_process; i++) {
39     position = i;
40     for (int j = i + 1; j < number_of_process; j++){
41         if (process[j].priority > process[position].priority)
42             position = j;
43     }
44     temp_process = process[i];
45     process[i] = process[position];
46     process[position] = temp_process;
47 }
48 process[0].waiting_time = 0;
49 for (int i = 1; i < number_of_process; i++) {
50     process[i].waiting_time = 0;
51     for (int j = 0; j < i; j++) {
52         process[i].waiting_time += process[j].burst_time;
53     }
54     total += process[i].waiting_time;
55 }
56 average_waiting_time = (float) total / (float) number_of_process;
57 total = 0;
58 printf("\n\nProcess_name \t Burst Time \t Waiting Time \t Turnaround Time\n");
59 printf("-----\n");
60 for (int i = 0; i < number_of_process; i++) {
61     process[i].turn_around_time = process[i].burst_time + process[i].waiting_time;
62     total += process[i].turn_around_time;
63     printf("\t %c \t\t %d \t\t %d \t\t %d", process[i].process_name, process[i].burst_time, process[i].waiting_time, process[i].turn_around_time);
64     printf("\n-----\n");
65 }
66 average_turnaround_time = (float) total / (float) number_of_process;
67 printf("\n\n Average Waiting Time : %f", average_waiting_time);
68 printf("\n Average Turnaround Time: %f\n", average_turnaround_time);
69 return 0;
70 }
71

```


Enter the total number of Processes: 2

Please Enter the Burst Time and Priority of each process:

Enter the details of the process A

Enter the burst time: 4

Enter the priority: 6

Enter the details of the process B

Enter the burst time: 3

Enter the priority: 2

Process_name	Burst Time	Waiting Time	Turnaround Time
A	4	0	4
B	3	4	7

Average Waiting Time : 2.000000

Average Turnaround Time: 5.500000

...Program finished with exit code 0

Press ENTER to exit console.