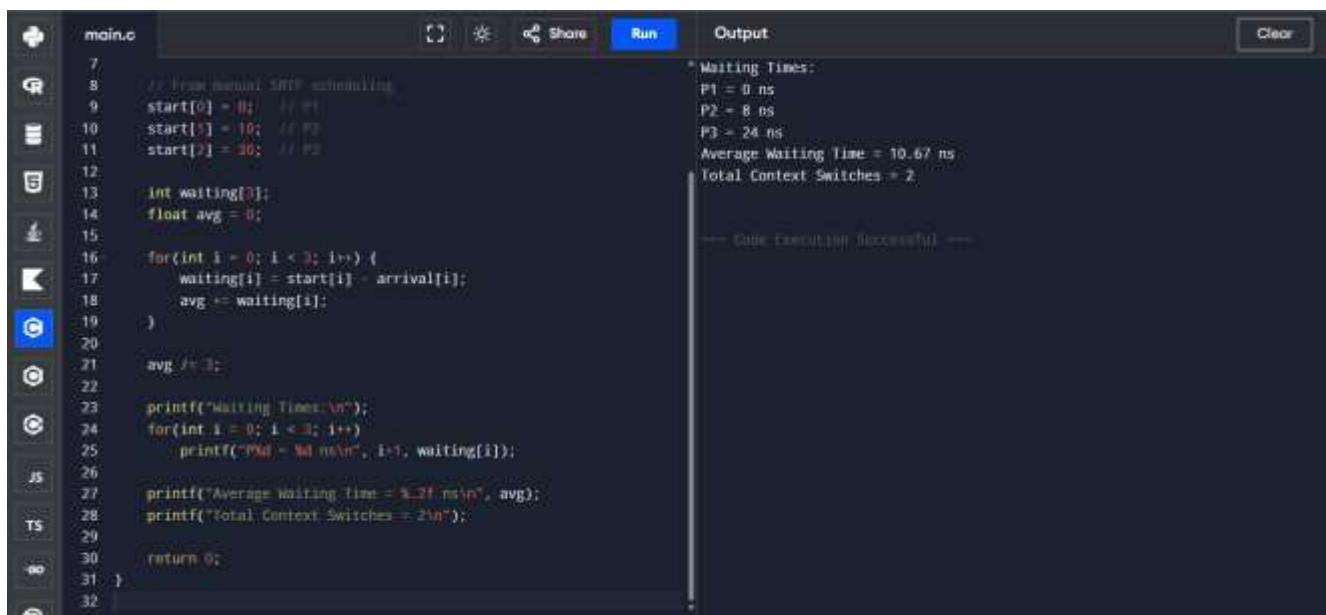


Practical 5

In an operating system three CPU-intensive processes are ready for execution, which require 10ns, 20ns and 30ns and arrival at times Qns, 2ns and 6ns, respectively. Write a program to calculate the total number of context switches needed if the operating system implements a shortest job first (preemptive) scheduling algorithm. Also calculate the average time for which the processes have to wait before getting the CPU.

Programme & Output:



The screenshot shows a code editor window with a dark theme. On the left is a sidebar with various icons for file operations. The main area contains a C program named 'main.c'. The code initializes three processes (P1, P2, P3) with start times at 0, 10, and 30 ns respectively. It calculates waiting times for each process based on their arrival times and the start of the next process. The output section displays the calculated waiting times, average waiting time, and total context switches.

```
main.c
7
8 // First come first serve scheduling
9 start[0] = 0; // P1
10 start[1] = 10; // P2
11 start[2] = 30; // P3
12
13 int waiting[3];
14 float avg = 0;
15
16 for(int i = 0; i < 3; i++) {
17     waiting[i] = start[i] - arrival[i];
18     avg += waiting[i];
19 }
20
21 avg /= 3;
22
23 printf("Waiting Times:\n");
24 for(int i = 0; i < 3; i++)
25     printf("%d = %d ns\n", i+1, waiting[i]);
26
27 printf("Average Waiting Time = %.2f ns\n", avg);
28 printf("Total Context Switches = 2\n");
29
30 return 0;
31 }
32
```

Output:

```
* Waiting Times:
P1 = 0 ns
P2 = 8 ns
P3 = 24 ns
Average Waiting Time = 10.67 ns
Total Context Switches = 2

--- Code Execution Successful ---
```