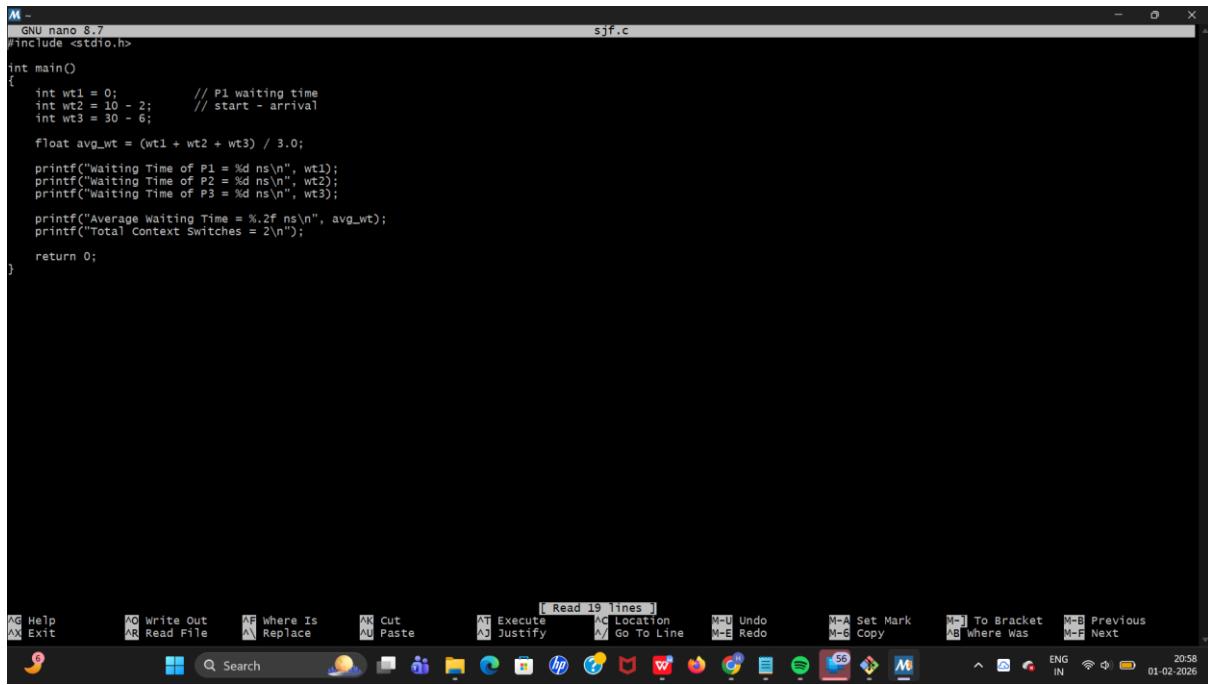


[Process management Based practical]

In an operating system three CPU-intensive processes are ready for execution, which require 10ns, 20ns and 30ns and arrival at times 0ns, 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.

INPUT:

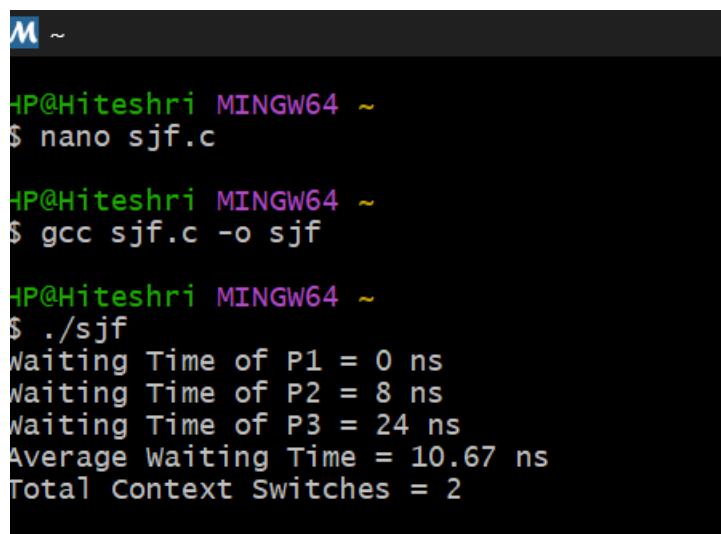


The screenshot shows a terminal window titled "GNU nano 8.7 sjf.c". The code in the editor is as follows:

```
GNU nano 8.7
#include <stdio.h>
int main()
{
    int wt1 = 0;           // P1 waiting time
    int wt2 = 10 - 2;     // start - arrival
    int wt3 = 30 - 6;
    float avg_wt = (wt1 + wt2 + wt3) / 3.0;
    printf("Waiting Time of P1 = %d ns\n", wt1);
    printf("Waiting Time of P2 = %d ns\n", wt2);
    printf("Waiting Time of P3 = %d ns\n", wt3);
    printf("Average Waiting Time = %.2f ns\n", avg_wt);
    printf("Total Context Switches = 2\n");
    return 0;
}
```

The terminal window has a dark background and light-colored text. The status bar at the bottom shows various icons and the date/time: "01-02-2026 20:58".

OUTPUT:



The screenshot shows a terminal window with the following session:

```
HP@Hiteshri MINGW64 ~
$ nano sjf.c

HP@Hiteshri MINGW64 ~
$ gcc sjf.c -o sjf

HP@Hiteshri MINGW64 ~
$ ./sjf
Waiting Time of P1 = 0 ns
Waiting Time of P2 = 8 ns
Waiting Time of P3 = 24 ns
Average Waiting Time = 10.67 ns
Total Context Switches = 2
```