In an operating system three CPU-intensive processes are ready for execution which require 10 ns, 20 ns and 30 ns and arrive at times 0 ns, 2 ns, and 6 ns, 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.
- The average waiting time of the processes before getting the CPU.

CODE:

```
  GNU nano 8.7                                                    sjf.c
#include <stdio.h>

int main() {

    int n = 3;

    int at[] = {0, 2, 6};      // Arrival times
    int bt[] = {10, 20, 30};   // Burst times

    // Completion times calculated using SRTF logic manually
    int ct[] = {10, 30, 60};

    int wt[3];
    int total_wt = 0;

    for(int i = 0; i < n; i++) {
        wt[i] = ct[i] - at[i] - bt[i];
        total_wt += wt[i];
    }

    printf("Process\tArrival\tBurst\tWaiting\n");

    for(int i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\n", i+1, at[i], bt[i], wt[i]);
    }

    printf("\nAverage Waiting Time = %.2f ns\n", (float)total_wt/n);
    printf("Total Context Switches = 2\n");

    return 0;
}
```

OUTPUT:

```
VIDISHA@DESKTOP-KHV52LE MINGW64 ~/OneDrive/Desktop
$ nano sjf.c

VIDISHA@DESKTOP-KHV52LE MINGW64 ~/OneDrive/Desktop
$ gcc sjf.c -o sjf

VIDISHA@DESKTOP-KHV52LE MINGW64 ~/OneDrive/Desktop
$ ./sjf
Process Arrival Burst   Waiting
P1      0       10      0
P2      2       20      8
P3      6       30      24

Average Waiting Time = 10.67 ns
Total Context Switches = 2
```