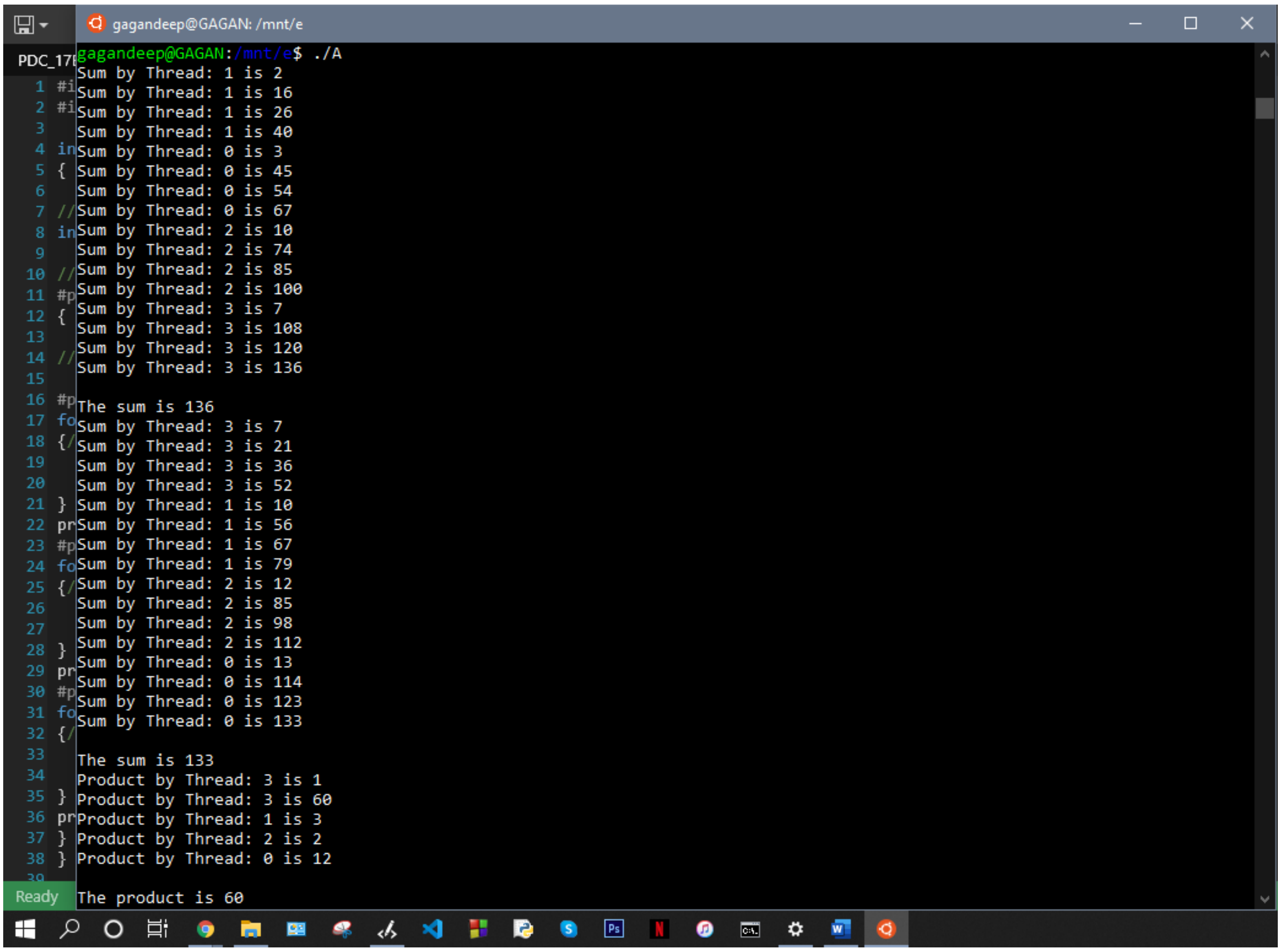
Dated : 26|08|2019
Registration No. : 17BCI0140
Name : Gagan Deep Singh
Aim: Write a simple OpenMP program to demonstrate the use of schedule clause

a. Statically assign the loop iterations to threads
b. Dynamically assign one iteration to each threads

SOURCE CODE:

```
#include <omp.h>
#include <stdio.h>

int main(void)
{

// Initialize the variables
int i, arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}, sum = 0, sum2 =
0, prod = 1;

//Pragma Functions
#pragma omp shared(arr, aum, prod) private(i)
{

// Define Static Scheduling

#pragma omp parallel for schedule(static, 1)
for (i = 0; i < 16; ++i)
{//Sum of ‘n’ Array Elements
        sum += arr[i];
        printf("Sum by Thread: %d is %d\n", omp_get_thread_num(), sum);
}
printf("\nThe sum is %d\n", sum);
#pragma omp parallel for schedule(static, 2)
for (i = 0; i < 16; ++i)
{//Product of ‘n’ Array Elements
        sum2 += arr[i];
        printf("Sum by Thread: %d is %d\n", omp_get_thread_num(), sum2);
}
printf("\nThe sum is %d\n", sum2);
#pragma omp parallel for schedule (dynamic, 1)
for (i = 0; i < 5; ++i)
{//Product of ‘n’ Array Elements
        prod *= arr[i];
        printf("Product by Thread: %d is %d\n", omp_get_thread_num(), prod);
}
printf("\nThe product is %d\n", prod);
}
}
```

Dated : 26|08|2019
Registration No. : 17BCI0140
Name : Gagan Deep Singh
EXECUTION:



```
gagandeep@GAGAN:/mnt/e$ ./A
Sum by Thread: 1 is 2
Sum by Thread: 1 is 16
Sum by Thread: 1 is 26
Sum by Thread: 1 is 40
Sum by Thread: 0 is 3
Sum by Thread: 0 is 45
Sum by Thread: 0 is 54
Sum by Thread: 0 is 67
Sum by Thread: 2 is 10
Sum by Thread: 2 is 74
Sum by Thread: 2 is 85
Sum by Thread: 2 is 100
Sum by Thread: 3 is 7
Sum by Thread: 3 is 108
Sum by Thread: 3 is 120
Sum by Thread: 3 is 136

The sum is 136
Sum by Thread: 3 is 7
Sum by Thread: 3 is 21
Sum by Thread: 3 is 36
Sum by Thread: 3 is 52
Sum by Thread: 1 is 10
Sum by Thread: 1 is 56
Sum by Thread: 1 is 67
Sum by Thread: 1 is 79
Sum by Thread: 2 is 12
Sum by Thread: 2 is 85
Sum by Thread: 2 is 98
Sum by Thread: 2 is 112
Sum by Thread: 0 is 13
Sum by Thread: 0 is 114
Sum by Thread: 0 is 123
Sum by Thread: 0 is 133

The sum is 133
Product by Thread: 3 is 1
Product by Thread: 3 is 60
Product by Thread: 1 is 3
Product by Thread: 2 is 2
Product by Thread: 0 is 12

The product is 60
```