VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Computer Science & Engineering

CSE4001

Parallel and Distributed Computing

LAB ASSIGNMENT 2

Submitted to **Prof. DEEBAK B.D.**

TOPIC: PROBLEMS USING OPENMP

NAME: PUNIT MIDDHA

REG.NO: 19BCE2060

SLOT: L55+L56

DATE: 31/08/2021

Write a simple OpenMP program to demonstrate the use of 'for' clause.

A - Print 'n' array elements

B - Sum of n' array elements

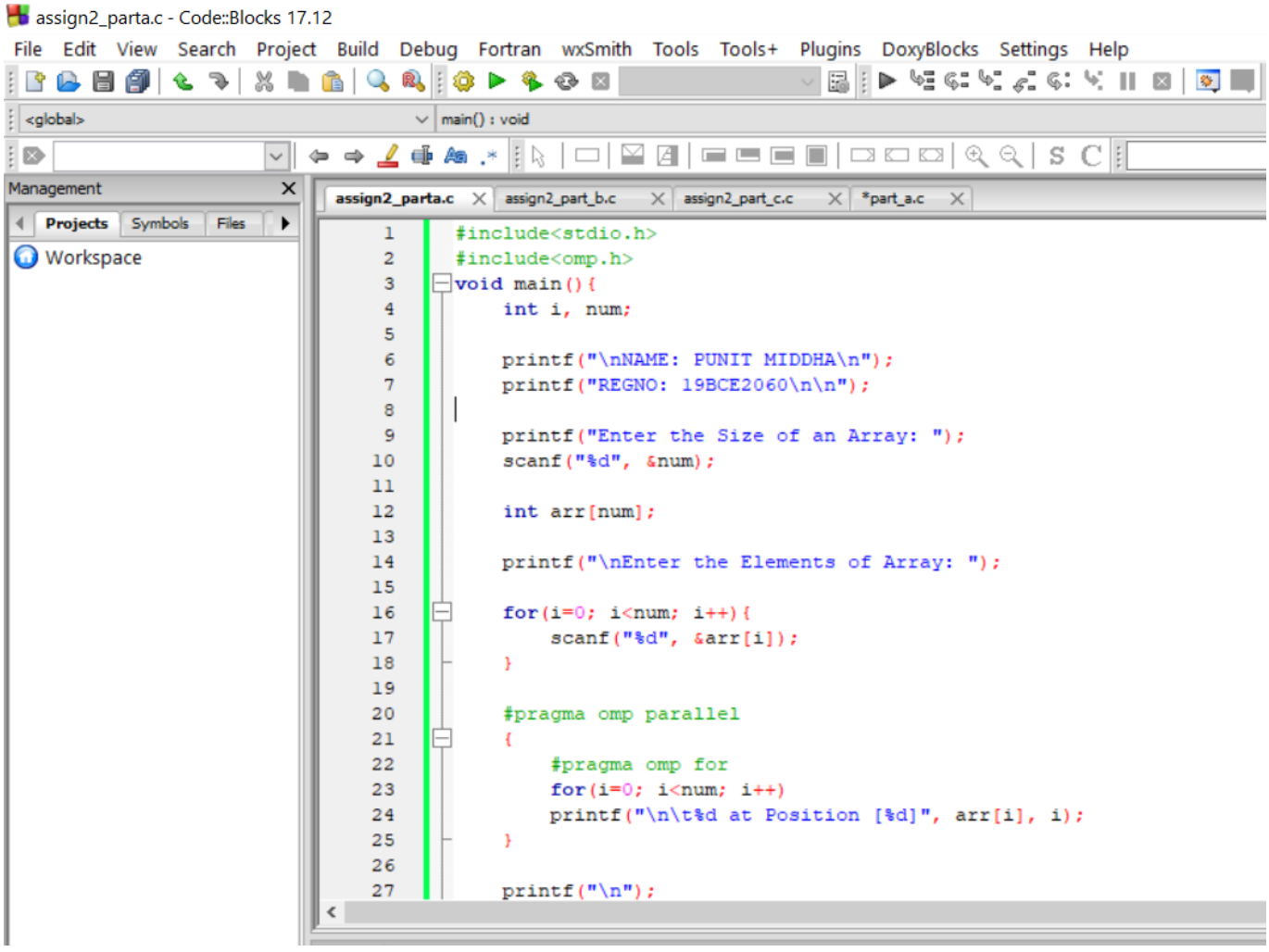
C - Product of n' array elements

PART-A:

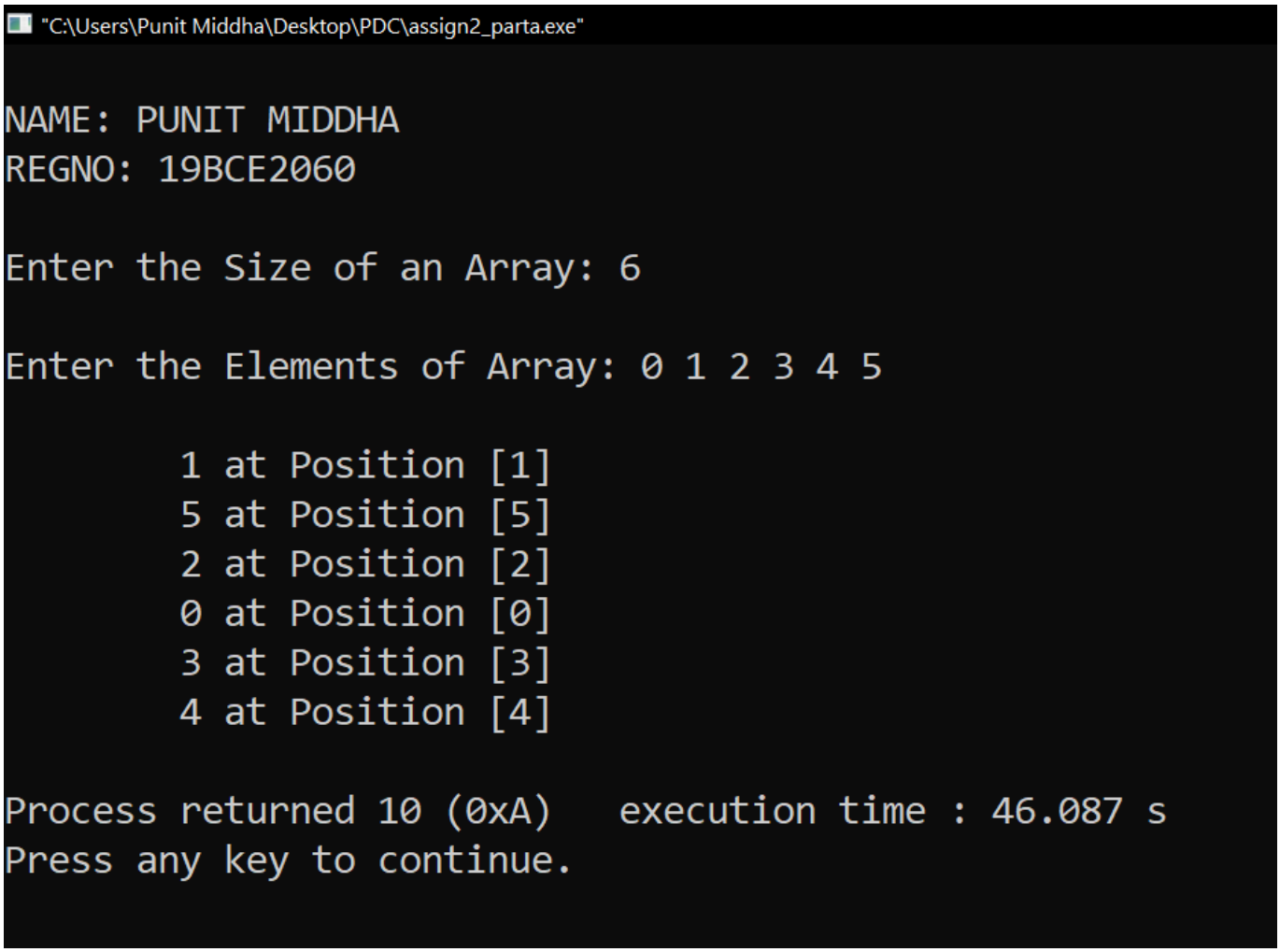
SOURCE CODE:

```
#include<stdio.h>
#include<omp.h>
void main(){
    int i, num;

    printf("\nNAME: PUNIT MIDDHA\n");
    printf("REGNO: 19BCE2060\n\n");

    printf("Enter the Size of an Array: ");
    scanf("%d", &num);

    int arr[num];

    printf("\nEnter the Elements of Array: ");

    for(i=0; i<num; i++){
        scanf("%d", &arr[i]);
    }

    #pragma omp parallel
```

```
    {
        #pragma omp for
        for(i=0; i<num; i++)
        printf("\n\t%d at Position [%d]", arr[i], i);
    }

    printf("\n");
}
```



EXECUTION:



REMARKS:

In (a) part the for clause is used with #pragma omp for. If we use it without it i.e., normal formal loop the program would not work.

PART-B:

SOURCE CODE:

```
#include<stdio.h>

#include<omp.h>

void main(){

    int i, num;
```

```
    printf("\nNAME: PUNIT MIDDHA\n");
    printf("REGNO: 19BCE2060\n\n");

    printf("Enter the Size of an Array: ");
    scanf("%d", &num);

    int array[num], sum=0;

    printf("\nEnter the Elements of Array: ");

    for(i=0; i<num; i++){
        scanf("%d", &array[i]);
    }

    #pragma omp parallel shared(sum,array) private(i)
    {
        #pragma omp for
        for(i=0; i<num; i++){
            sum+=array[i];
            printf("\nSum at %d th Position = %d", i, sum);
        }
    }
    printf("\n");
}
```

assign2_part_b.c - Code::Blocks 17.12

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : void

Management

Projects Symbols Files

Workspace

assign2_parta.c | assign2_part_b.c | assign2_part_c.c | *part_a.c

```c
#include<stdio.h>
#include<omp.h>
void main(){
    int i, num;

    printf("\nNAME: PUNIT MIDDHA\n");
    printf("REGNO: 19BCE2060\n\n");

    printf("Enter the Size of an Array: ");
    scanf("%d", &num);

    int array[num], sum=0;

    printf("\nEnter the Elements of Array: ");

    for(i=0; i<num; i++){
        scanf("%d", &array[i]);
    }

    #pragma omp parallel shared(sum,array) private(i)
    {
        #pragma omp for
        for(i=0; i<num; i++){
            sum+=array[i];
            printf("\nSum at %d th Position = %d", i, sum);
        }
    }
```

EXECUTION:

Select "C:\Users\Punit Middha\Desktop\PDC\assign2_part_b.exe"

```
NAME: PUNIT MIDDHA
REGNO: 19BCE2060

Enter the Size of an Array: 5

Enter the Elements of Array: 10 20 30 40 50

Sum at 1 th Position = 20
Sum at 0 th Position = 30
Sum at 3 th Position = 120
Sum at 4 th Position = 80
Sum at 2 th Position = 150

Process returned 10 (0xA)   execution time : 7.637 s
Press any key to continue.
```

REMARKS:

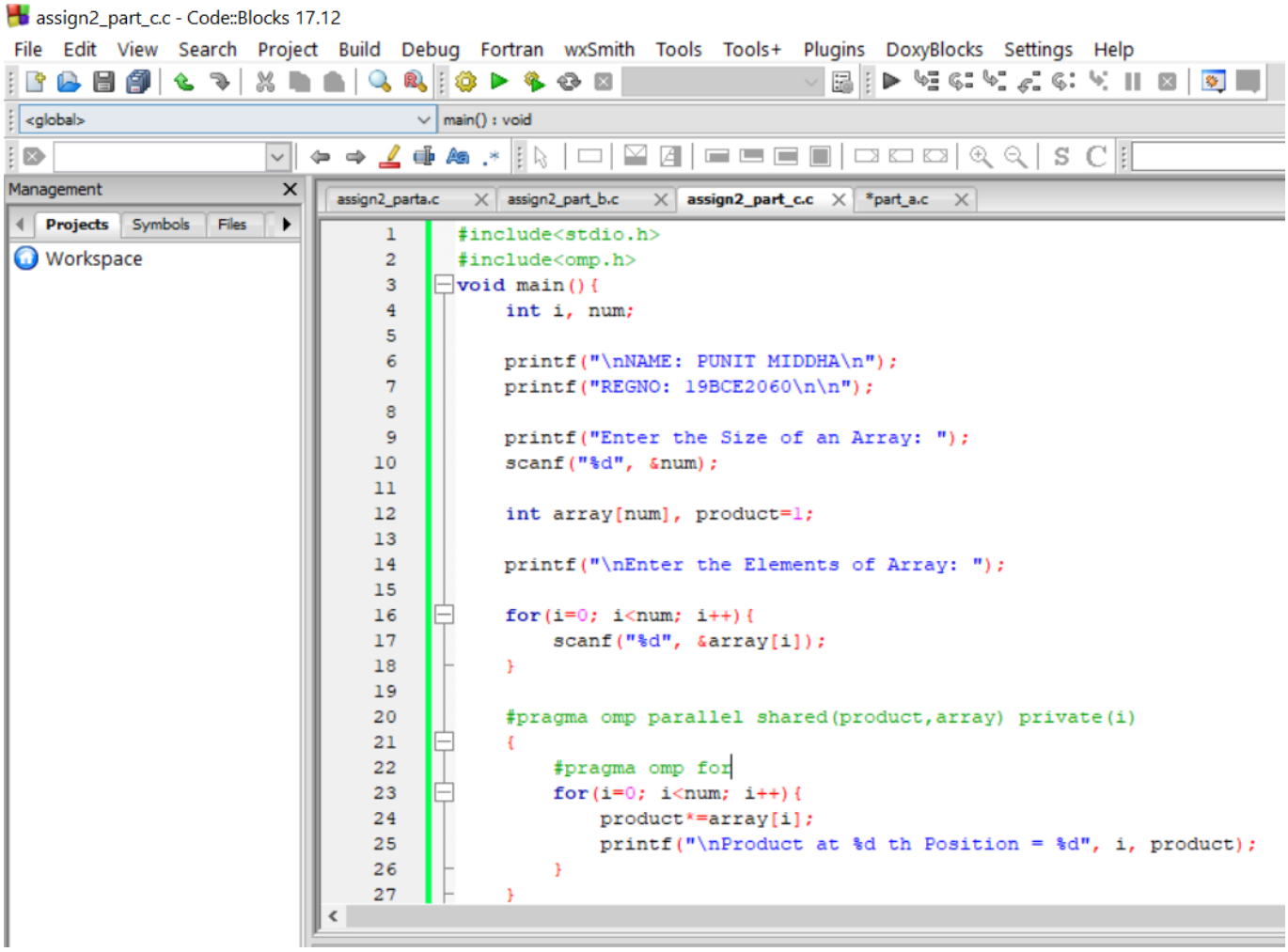
In (b) part I needed to print the sum of all the elements in the array. For this I have used shared(sum,array) and private(i) clause. Array and sum are shared elements while i is private. This will give us the correct sum of elements. #pragma omp for is used to run the "for" loop inside the #pragma omp parallel.

PART-C:

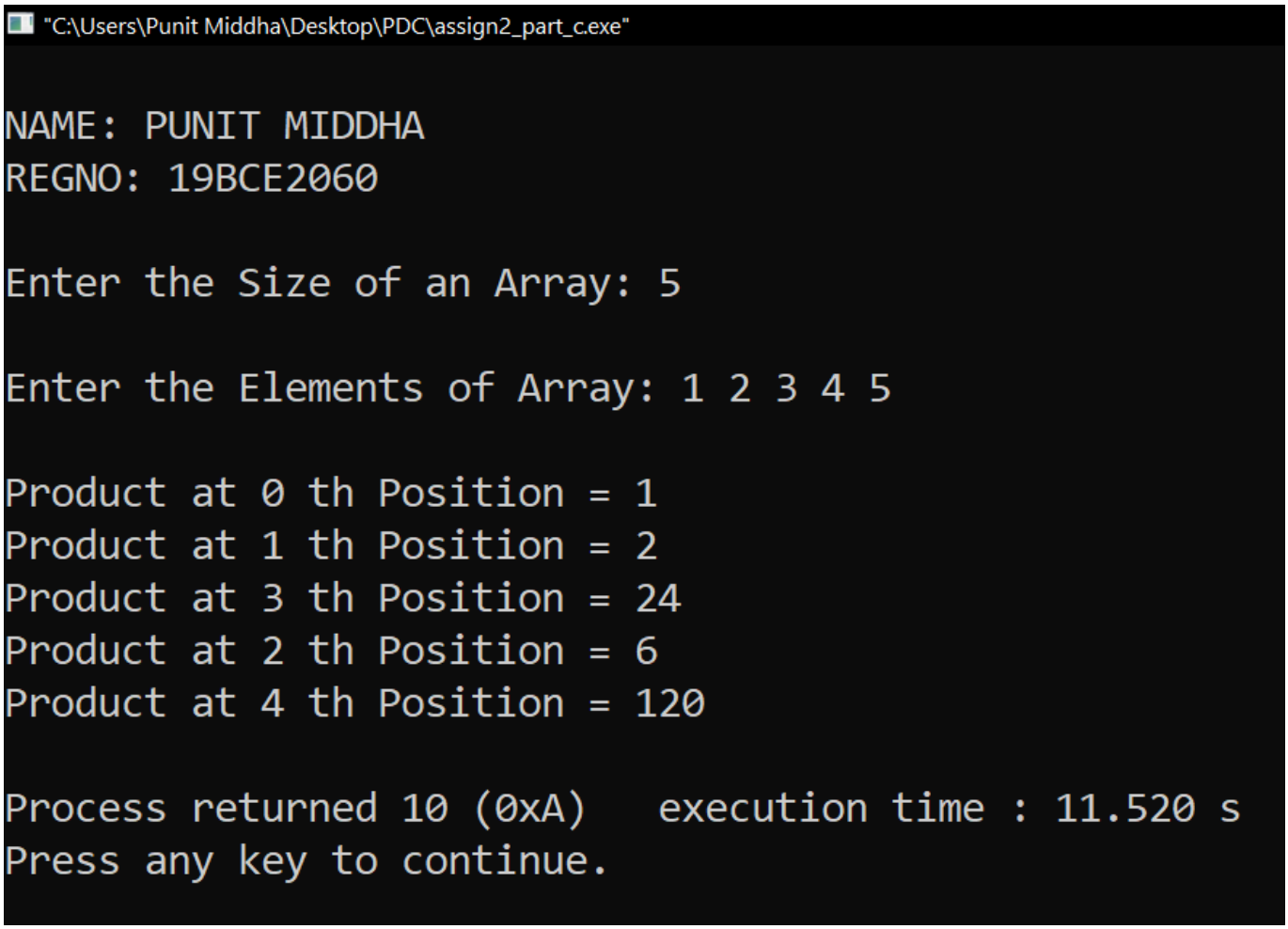
SOURCE CODE:

```
#include<stdio.h>

#include<omp.h>

void main(){

    int i, num;


    printf("\nNAME: PUNIT MIDDHA\n");

    printf("REGNO: 19BCE2060\n\n");


    printf("Enter the Size of an Array: ");

    scanf("%d", &num);


    int array[num], product=1;


    printf("\nEnter the Elements of Array: ");


    for(i=0; i<num; i++){

        scanf("%d", &array[i]);

    }
```

```
    #pragma omp parallel shared(product,array) private(i)
    {
        #pragma omp for
        for(i=0; i<num; i++){
            product*=array[i];
            printf("\nProduct at %d th Position = %d", i, product);
        }
    }

    printf("\n");
}
```



EXECUTION:



REMARKS:

In (c) part I needed to print the product of all the elements in the array. For this I have used shared(product,array) and private(i) clause. Array and product are shared elements while i is private. This will give us the correct product of elements.