Dated: 27/07/2020 Assessment No. : 2
18BCE0660 SYED AYAZ IMAM

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

PARALLEL & DISTRIBUTED COMPUTING(L31+32)

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

- Print *'n'* array elements
- Sum of *n'* array elements
- Product of *n'* array elements

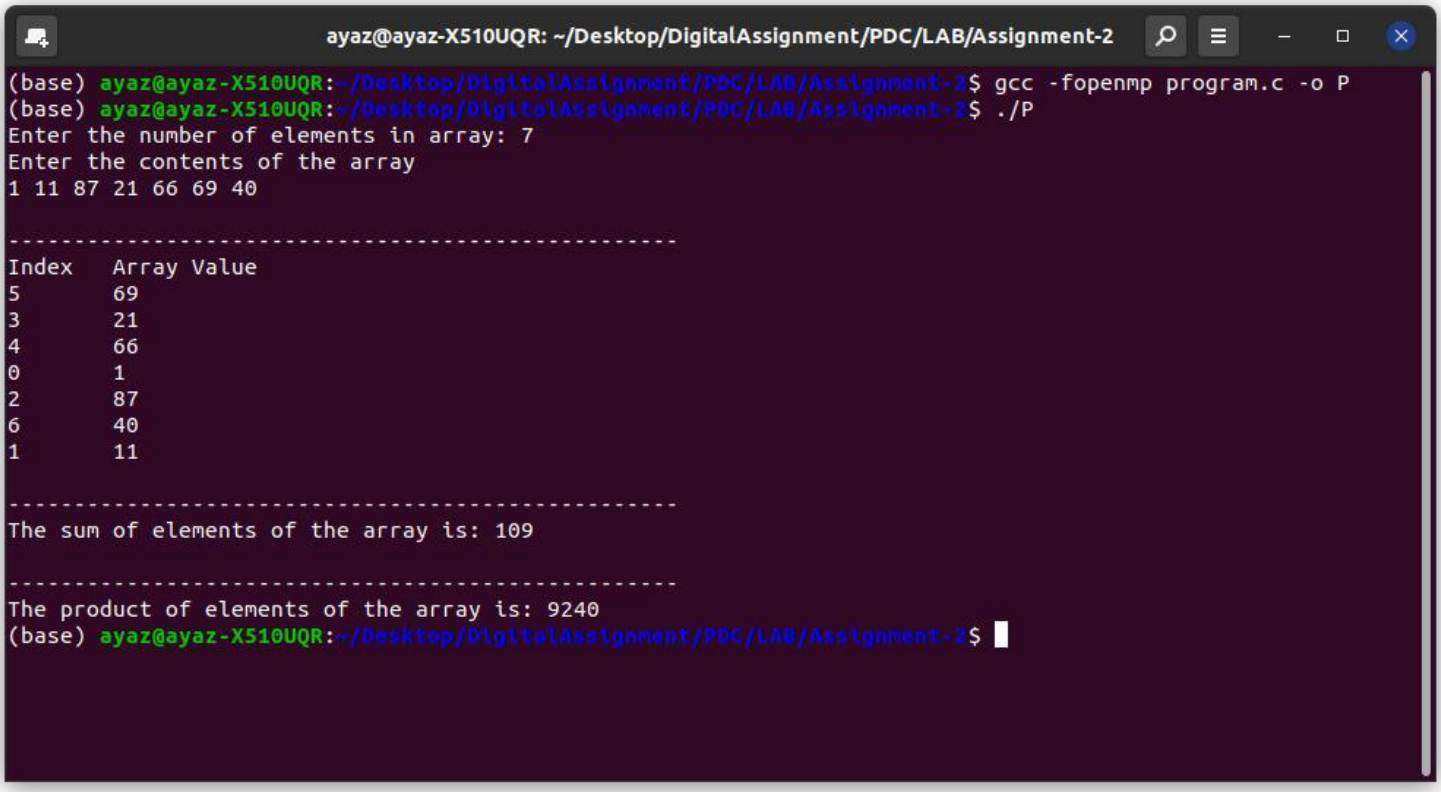
SOURCE CODE:

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

void printArray(int arr[], int n);
void sumArray(int arr[], int n);
void productArray(int arr[], int n);

void main()
{
    int i, n;
    //Populating the array
    printf("Enter the number of elements in array: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter the contents of the array\n");
    for(i = 0; i<n; i++){
        scanf("%d", &arr[i]);
    }
```

```
    printArray(arr, n);
    sumArray(arr, n);
    productArray(arr, n);
}

void printArray(int arr[], int n)
{
    printf("\n-----------------------------------------------------\n");
    printf("Index\tArray Value\n");
    #pragma omp parallel
    {
    #pragma omp for
    for(int i=0; i<n; i++)
    printf("%d\t%d\n", i, arr[i]);
    }
}

void sumArray(int arr[], int n)
{
    int sum = 0;
    printf("\n-----------------------------------------------------\n");
    #pragma omp parallel
    {
        #pragma omp for
        for(int i= 0; i<n; i++)
        {
            sum += arr[i];
        }
    }
    printf("The sum of elements of the array is: %d\n", sum);
}
```

```
void productArray(int arr[], int n)

{

    int prod = 1;

    printf("\n-------------------------------------------------\n");

    #pragma omp parallel

    {

        #pragma omp for

        for(int i= 0; i<n; i++)

        {

            prod *= arr[i];

        }

    }

    printf("The product of elements of the array is: %d\n", prod);

}
```

EXECUTION:



Dated: 27/07/2020 Assessment No. : 2
18BCE0660 SYED AYAZ IMAM

REMARKS:

From the given assessment was to explore the basic concepts of ‘for’ clause parallel programming by printing, summing array contents and producing product of array.

- The printing of array is handled by ‘printArray’ function in the code.
- The summation of array is handled by ‘sumArray’ function in the code.
- The product of elements in array is produced by the ‘productArray’ function in the code.

#pragma omp parallel: The code under this syntax cell is forked into subprocesses which is handled by multiple threads of the processor.

#pragma omp for: This syntax is used to invoke paralellised approch to for loop.