Fall Sem 2021-22

Assignment: 2

Date : 28/08/21

Name : Ishan Sagar Jogalekar

Reg no : 19BCE2250

Course: Parallel and distributed computing LAB - CSE4001

Slot : L55+L56

Aim:

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.

Ans –
1. Print ‘n’ array elements :

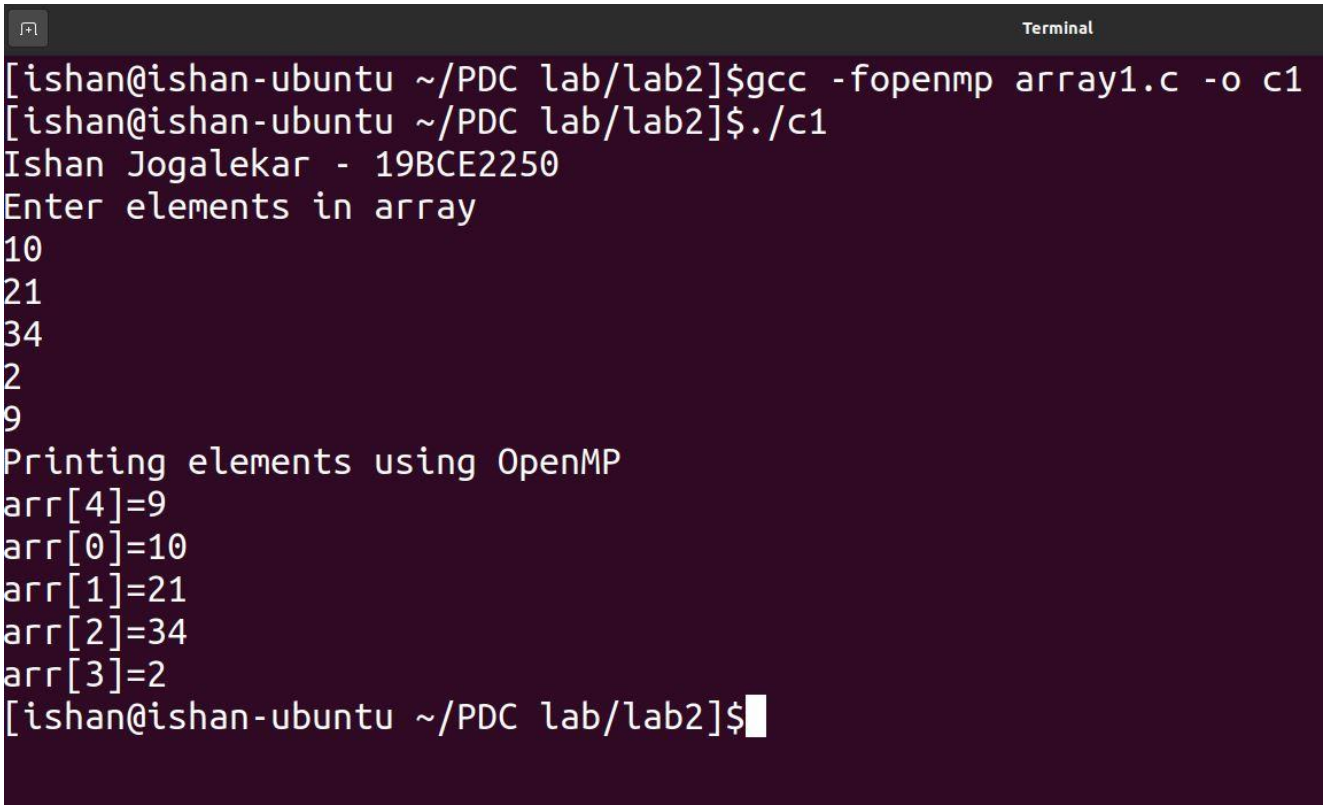
Source code:

```
#include<stdio.h>
#include<omp.h>
#include <stdlib.h>
/* Main Program */
void main()
{
   printf("Ishan Jogalekar - 19BCE2250\n");

   int arr[5];
   printf("Enter elements in array\n");
   // taking input and storing it in an array
   for(int i = 0; i < 5; ++i) {
         scanf("%d", &arr[i]);
```

```
 }
printf("Printing elements using OpenMP\n");
/* OpenMP Parallel For With Reduction Clause */
#pragma omp parallel
{
        #pragma omp for
          for(int i=0;i<5;i++)
                {
                        printf("arr[%d]=%d\n",i,arr[i]);
                }
        }
}
```

Execution:



Result:

1. The header file to be included for this experiment is “omp.h” which isOpenMP provided header file. It provides functions for parallel programming.

2. In main() function #pragma omp parallel is tagged used to start OpenMP parallel block. int arr[] is used to initialize array.

3. for loop and scanf() is used before #pragma omp parallel to take user input for elements in array.

4. #pragma omp for is used for loop parallelism.

5. #pragma omp parallel for &

   #pragma omp parallel
   {
   #pragma omp for both are similar functions.

6. Inside #pragma omp parallel **for loop** is initialized to print array elements.

2. Sum of n' array elements –

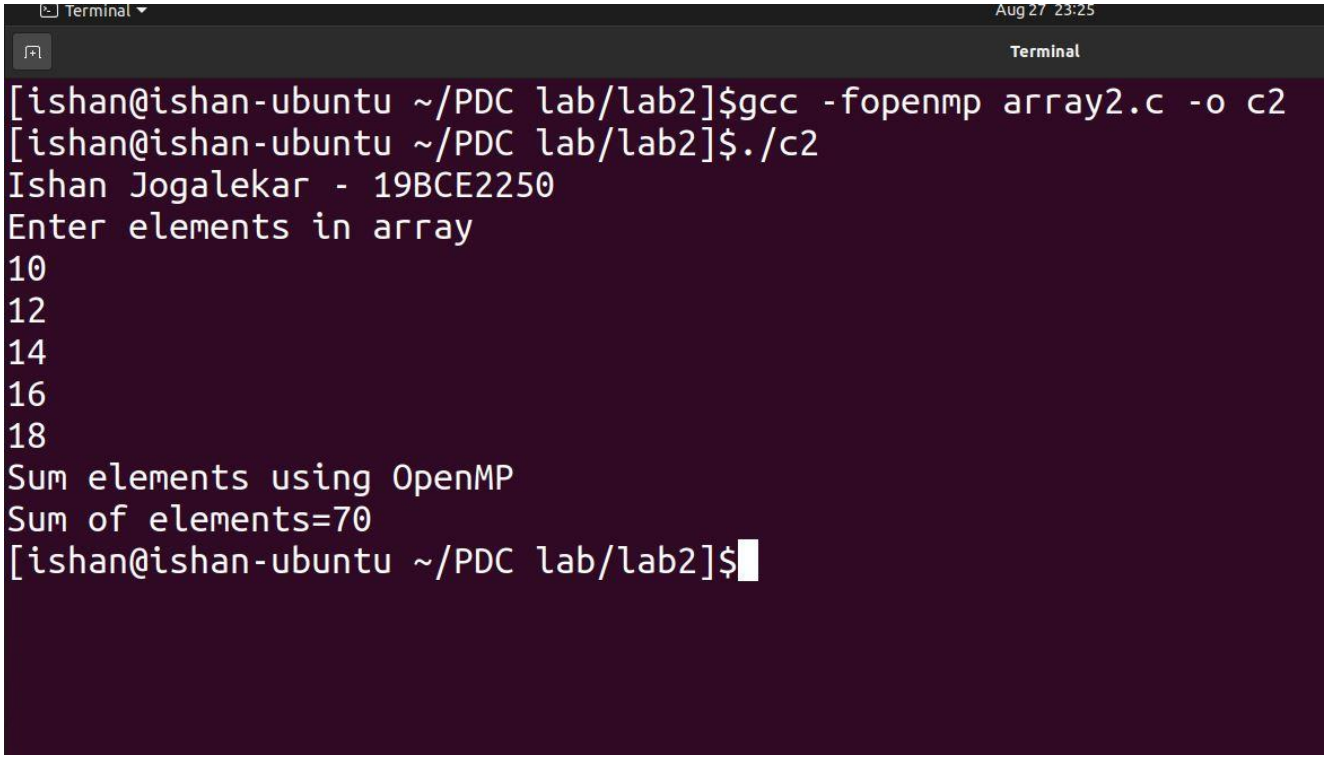
Source code:

```
#include<stdio.h>
#include<omp.h>
#include <stdlib.h>
void main()
{
  printf("Ishan Jogalekar - 19BCE2250\n");
  int arr[5];
  int sum=0; //Sum as addition

        printf("Enter elements in array\n");
        // taking input and storing it in an array
  for(int i = 0; i < 5; ++i) {
    scanf("%d", &arr[i]);
         }
         printf("Sum elements using OpenMP\n");
```

```
/* OpenMP Parallel For With Reduction Clause */
#pragma omp parallel for
  for (int i=0;i<5;i++){
    sum+=arr[i];
  }
  printf("Sum of elements=%d\n",sum);
}
```

Execution:



Result:

1. pragma omp parallel for is OpenMP parallel for with reduction clause.

2. Using it for loop is iteration. Inside parallel block loop is used to perform iteration in array elements.

3. sum+=arr[i]; is main function which performs addition within array elements. Int sum is declared in main() function only.

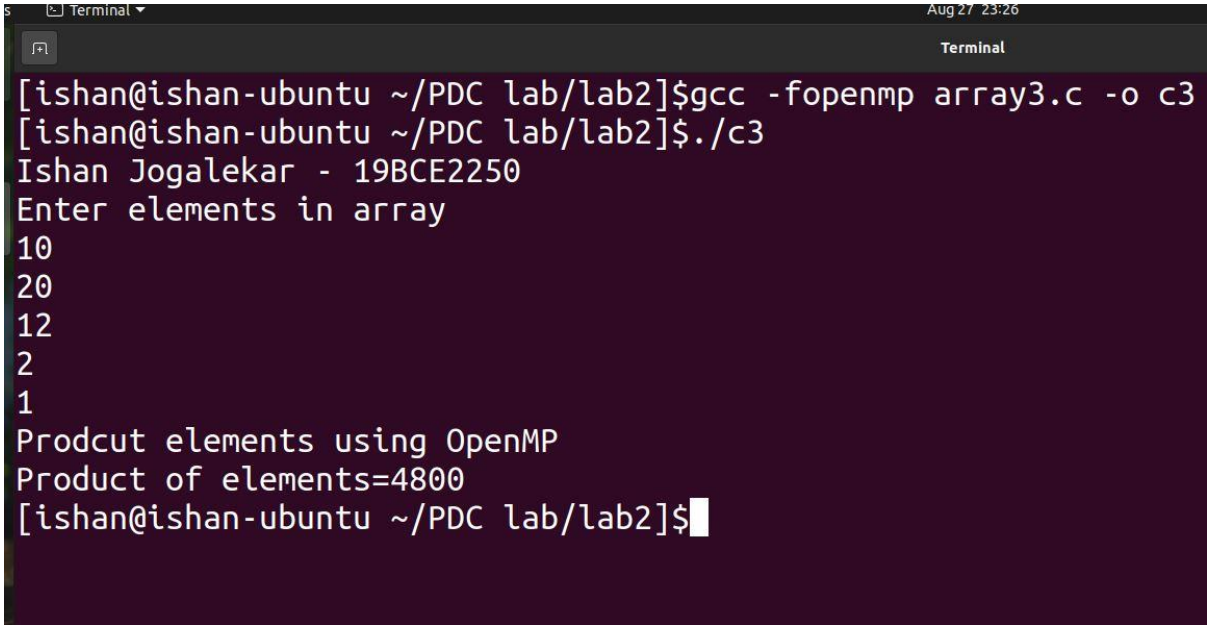
3. Product of n’ array elements –

Source code:

```
#include<stdio.h>
#include<omp.h>
#include <stdlib.h>
void main(){
  printf("Ishan Jogalekar - 19BCE2250\n");
  int arr[5];
  int result=1; //Result as product

    printf("Enter elements in array\n");
    // taking input and storing it in an array
  for(int i = 0; i < 5; ++i) {
     scanf("%d", &arr[i]);
     }
     printf("Prodcut elements using OpenMP\n");
   /* OpenMP Parallel For With Reduction Clause */
   #pragma omp parallel for
     for (int i=0;i<5;i++){
       result *= arr[i]; //Multipying elements and storing it in result
     }
     printf("Product of elements=%d\n",result);
   }
```

Execution:



Result:

1. pragma omp parallel for is OpenMP parallel for with reduction clause.

2. Using it for loop is iteration. Inside parallel block loop is used to perform iteration in array elements.

3. result *= arr[i]; is main function which performs multiplication within array elements. Int result is declared in main() function only.