

Parallel and Distributed Computing
CSE4001
Fall Semester 2020-21

Lab Assignment 3

ISHAAN OHRI
18BCE0265

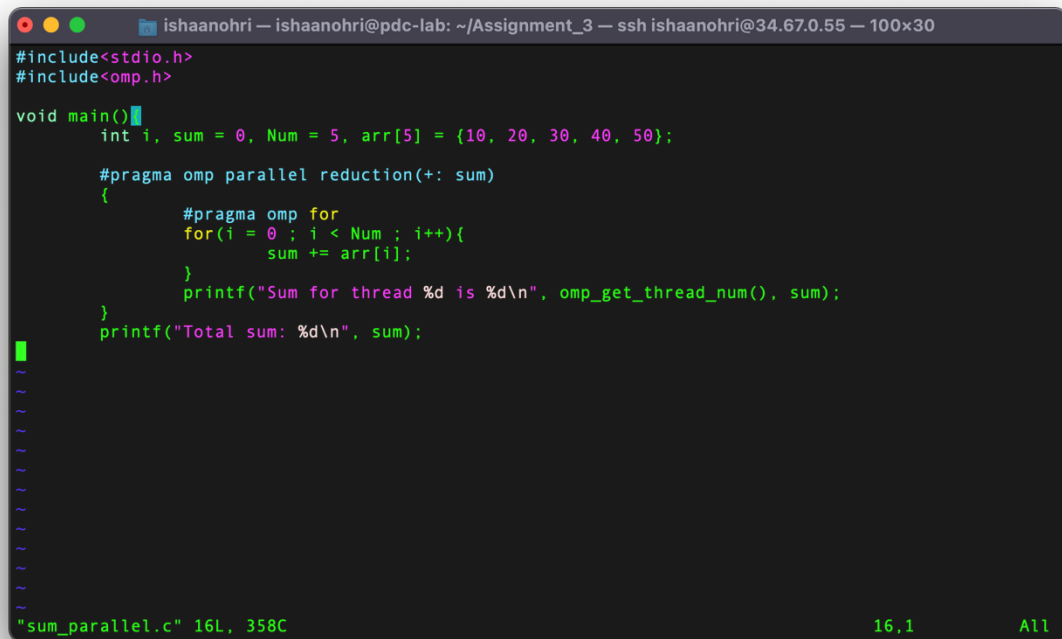
Aim:

Write a simple OpenMP program to demonstrate the use of reduction and critical clause

- Sum of 'n' array Using Reduction Clause
- Product of 'n' array using Reduction Clause
- Show a suitable example for Critical Clause

sum using reduction clause

Source Code:



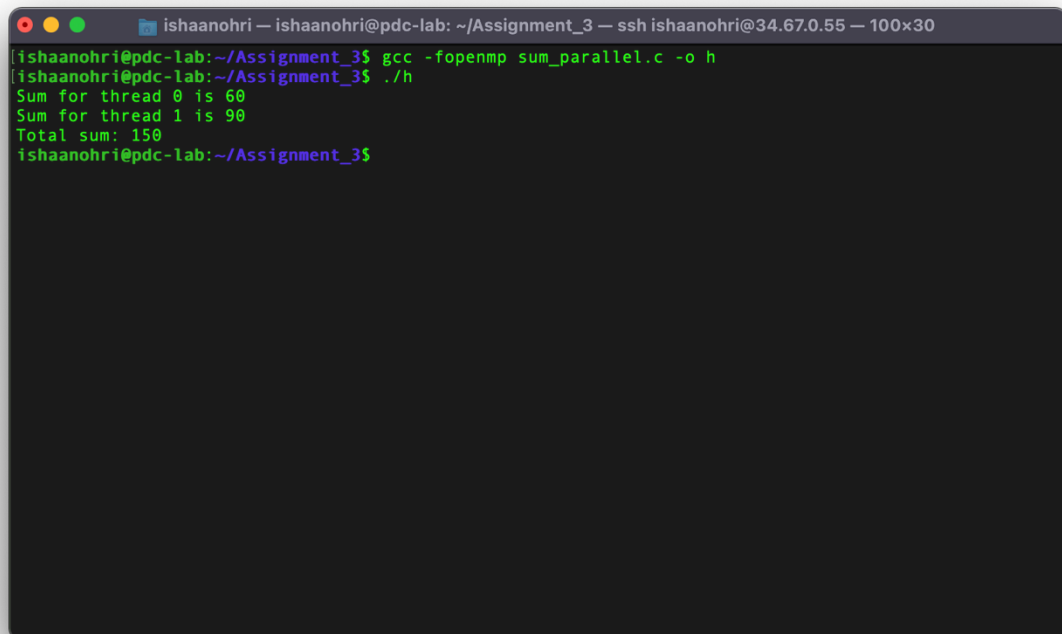
```
isshaanohri — isshaanohri@pdc-lab: ~/Assignment_3 — ssh isshaanohri@34.67.0.55 — 100x30
#include<stdio.h>
#include<omp.h>

void main()
{
    int i, sum = 0, Num = 5, arr[5] = {10, 20, 30, 40, 50};

    #pragma omp parallel reduction(+: sum)
    {
        #pragma omp for
        for(i = 0 ; i < Num ; i++){
            sum += arr[i];
        }
        printf("Sum for thread %d is %d\n", omp_get_thread_num(), sum);
    }
    printf("Total sum: %d\n", sum);
}

"sum_parallel.c" 16L, 358C 16,1 All
```

Execution:



```
isshaanohri — isshaanohri@pdc-lab: ~/Assignment_3 — ssh isshaanohri@34.67.0.55 — 100x30
[isshaanohri@pdc-lab:~/Assignment_3$ gcc -fopenmp sum_parallel.c -o h
[isshaanohri@pdc-lab:~/Assignment_3$ ./h
Sum for thread 0 is 60
Sum for thread 1 is 90
Total sum: 150
isshaanohri@pdc-lab:~/Assignment_3$
```

product using reduction clause

Source Code:

```

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

void main(){
    int i, prod = 1, Num = 5, arr[5] = {10, 20, 30, 40, 50};

    #pragma omp parallel reduction(*: prod)
    {
        #pragma omp for
        for(i = 0 ; i < Num ; i++){
            prod *= arr[i];
        }
        printf("Product for thread %d is %d\n", omp_get_thread_num(), prod);
    }
    printf("Total product: %d\n", prod);
}

```

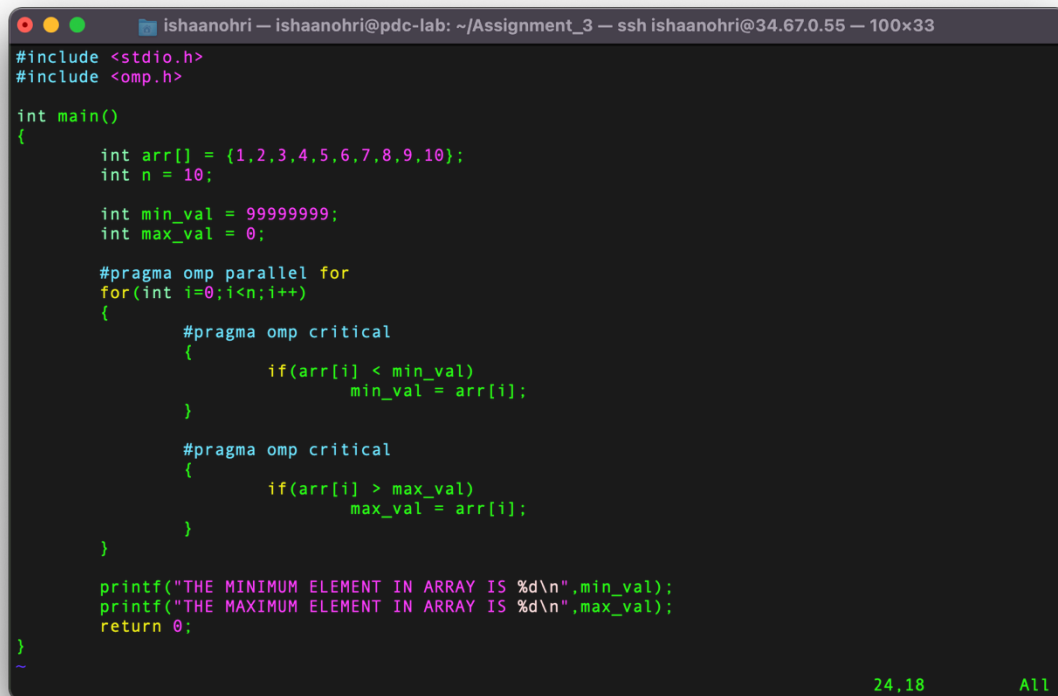
"product_parallel.c" 16L, 378C 13,38-52 All

Execution:

```
ishaanohri — ishaanohri@pdc-lab: ~/Assignment_3 — ssh ishaanohri@34.67.0.55 — 100x30
[ishaanohri@pdc-lab:~/Assignment_3$ gcc -fopenmp product_parallel.c -o h
[ishaanohri@pdc-lab:~/Assignment_3$ ./h
Product for thread 0 is 6000
Product for thread 1 is 2000
Total product: 12000000
ishaanohri@pdc-lab:~/Assignment_3$
```

example of critical clause

Source Code:



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

int main()
{
    int arr[] = {1,2,3,4,5,6,7,8,9,10};
    int n = 10;

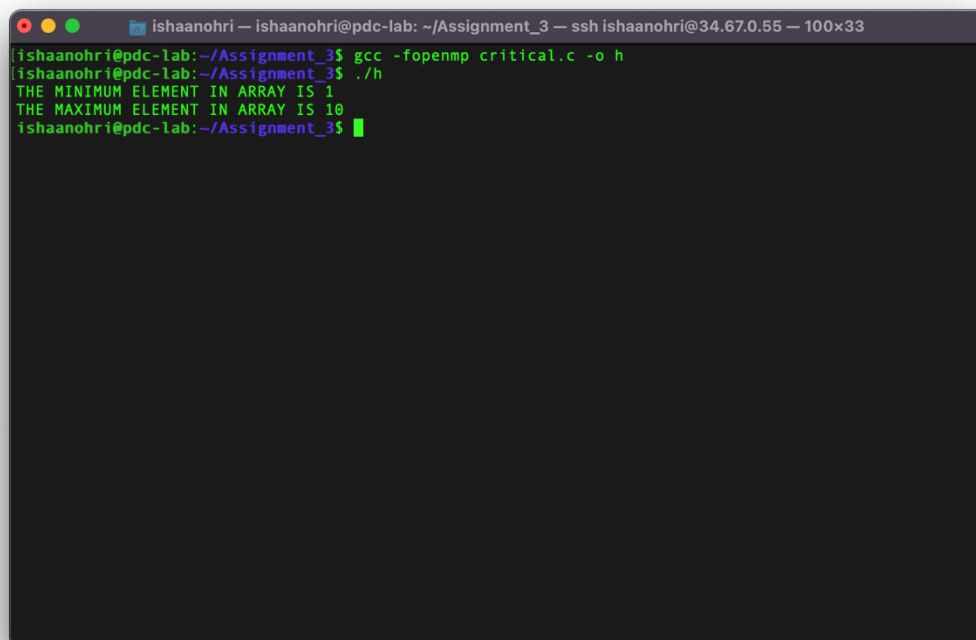
    int min_val = 99999999;
    int max_val = 0;

    #pragma omp parallel for
    for(int i=0;i<n;i++)
    {
        #pragma omp critical
        {
            if(arr[i] < min_val)
                min_val = arr[i];
        }

        #pragma omp critical
        {
            if(arr[i] > max_val)
                max_val = arr[i];
        }
    }

    printf("THE MINIMUM ELEMENT IN ARRAY IS %d\n",min_val);
    printf("THE MAXIMUM ELEMENT IN ARRAY IS %d\n",max_val);
    return 0;
}
```

Execution:



```
ishaanohri@pdc-lab: ~/Assignment_3
[ishaanohri@pdc-lab:~/Assignment_3$ gcc -fopenmp critical.c -o h]
[ishaanohri@pdc-lab:~/Assignment_3$ ./h]
THE MINIMUM ELEMENT IN ARRAY IS 1
THE MAXIMUM ELEMENT IN ARRAY IS 10
ishaanohri@pdc-lab:~/Assignment_3$
```

Result:

From this experiment I understood the usage of *reduction clause* using parallel pragma and arrays by sum of array elements, and product of array elements. Along with the use of *reduction clause*, I even understood the usage of *critical clause* by finding the minimum and the maximum element in the array. The above experiment was conducted and all results along with the source code have been attached above in the document. The experiment was assisted by Dr Deepak. I thank sir for his assistance.