

Name: Varun Menon

Course Code: CSE4001

Reg No: 19BCE1438

Faculty: Dr. Harini S

Lab Experiment 2

1. Array Addition using Parallel For

Code:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <omp.h>
#include <sched.h>
int main() {
    int a[10], b[10], c[10];
    int i;
    printf("Enter values of a array and b array\n");
    for(i = 0; i < 10; i++)
    {
        scanf("%d %d", &a[i], &b[i]);
    }
    #pragma omp parallel for
    for (i = 0; i < 10; i++)
    {
        c[i] = a[i] + b[i];
        printf("Thread %d\tValue %d\n", omp_get_thread_num(), c[i]);
    }
    printf("Values of c array\n");
    for(i = 0; i < 10; i++)
    {
        printf("%d\n", c[i]);
    }
    return 0;
}
```

Output:

```
C:\Users\menon\Desktop>gcc -fopenmp arrays_parallelfor.cpp

C:\Users\menon\Desktop>a.exe
Enter values of a array and b array
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
Thread 2      Value 19
Thread 2      Value 23
Thread 0      Value 3
Thread 0      Value 7
Thread 4      Value 35
Thread 5      Value 39
Thread 1      Value 11
Thread 1      Value 15
Thread 3      Value 27
Thread 3      Value 31
Values of c array
3
7
11
15
19
23
27
31
35
39

C:\Users\menon\Desktop>_
```

2. Sample for Private Variable

Code:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <omp.h>
int main()
{
    int numThreads, tid;
    #pragma omp parallel private(tid)
    {
        tid = omp_get_thread_num();
        printf("This is %d\n", tid);
        if(tid == 0)
        {
            numThreads = omp_get_num_threads();
            printf("Number of threads = %d\n", numThreads);
        }
    }
    return 0;
}
```

Output:

```
C:\Users\menon\Desktop>gcc -fopenmp private.cpp

C:\Users\menon\Desktop>a.exe
This is 3
This is 0
Number of threads = 6
This is 4
This is 5
This is 2
This is 1

C:\Users\menon\Desktop>
```

3. Array addition using parallel for with a, b, c as private arrays

Code:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <omp.h>
#include <sched.h>
int main() {
    int a[10], b[10], c[10];
    int i;
    printf("Enter values of a array and b array\n");
    for(i = 0; i < 10; i++)
    {
        scanf("%d %d", &a[i], &b[i]);
    }
    #pragma omp parallel for private(a,b,c)
    for (i = 0; i < 10; i++)
    {
        c[i] = a[i] + b[i];
        printf("Thread %d\tValue %d\n", omp_get_thread_num(), c[i]);
    }
    printf("Values of c array\n");
    for(i = 0; i < 10; i++)
    {
        printf("%d\n", c[i]);
    }
    return 0;
}
```

Output:

```
C:\Users\menon\Desktop>gcc -fopenmp array_private.cpp

C:\Users\menon\Desktop>a.exe
Enter values of a array and b array
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
Thread 2      Value 6421672
Thread 2      Value 35065768
Thread 1      Value 2031619
Thread 1      Value 32964364
Thread 5      Value 15405409
Thread 4      Value 1
Thread 0      Value 1669484792
Thread 0      Value -2147483625
Thread 3      Value 30810796
Thread 3      Value 30810792
Values of c array
0
1993375087
15408900
6422204
1993441701
1993765660
6422268
1993438845
8
1993374893

C:\Users\menon\Desktop>_
```

The values that are thrown up are garbage values as the array variables are private or local to each thread. This means that the values of c for which values are printed are not updated as they belong to a different thread.

4. Parallelize addition and subtraction of two integer variables a and b

Code:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <omp.h>
int main()
{
    int a, b, sum, difference, thNum;
    printf("Enter values of a and b: ");
    scanf("%d %d", &a, &b);
    #pragma omp parallel shared(a,b) private(sum, difference, thNum)
    {
        thNum = omp_get_thread_num();
        if(thNum%2)
        {
            sum = a+b;
            printf("Thread num %d; Sum = %d\n", id, sum);
        }
        else
        {
            difference = a-b;
            printf("Thread num %d; Difference = %d\n", id, difference);
        }
    }
}
```

Output:

```
C:\Users\menon\Desktop>gcc -fopenmp parallel_add_sub.cpp

C:\Users\menon\Desktop>a.exe
Enter values of a and b: 67 23
Thread num 1; Sum = 90
Thread num 2; Difference = 44
Thread num 0; Difference = 44
Thread num 3; Sum = 90
Thread num 4; Difference = 44
Thread num 5; Sum = 90

C:\Users\menon\Desktop>
```