**Name: Varun Menon**

**Reg No: 19BCE1438**

**Course Code: CSE4001**

**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++)

    {

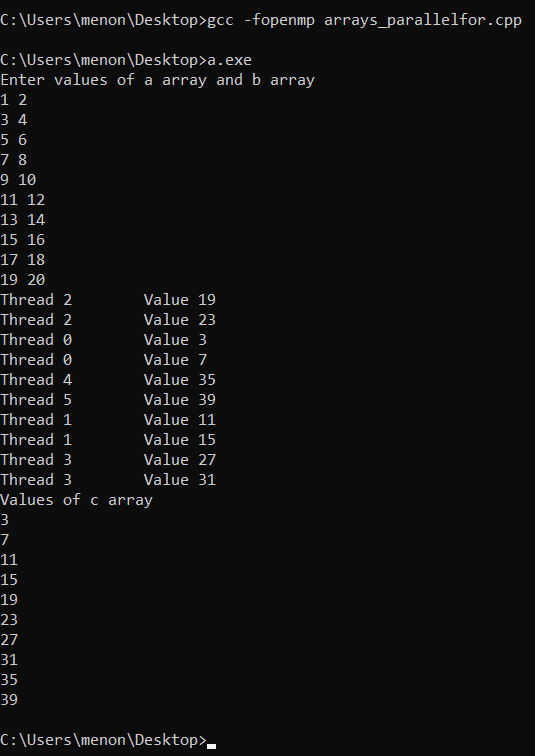
        printf("%d\n", c[i]);

    }

    return 0;

}

Output:



1. 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);

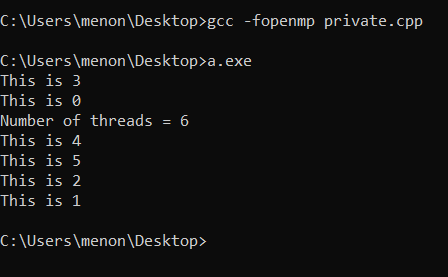
        }

    }

return 0;

}

Output:



1. 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++)

    {

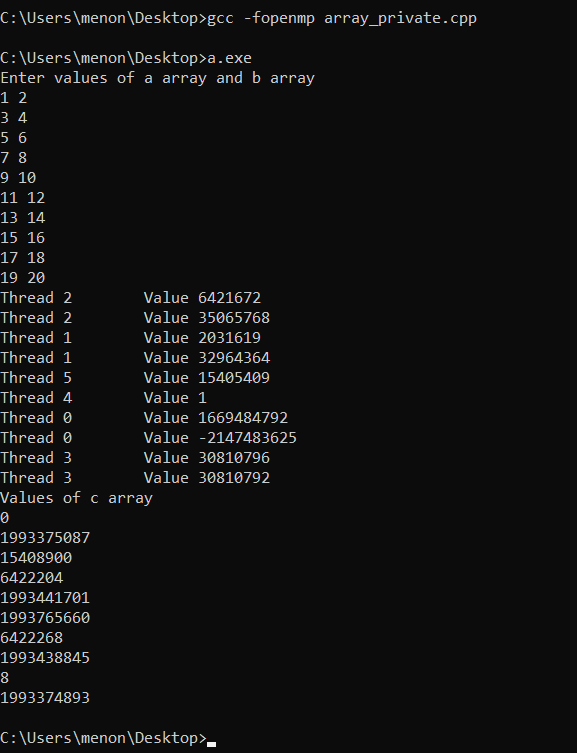
        printf("%d\n", c[i]);

    }

return 0;

}

Output:



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.

1. 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:

