**Name: Varun Menon**

**Reg No: 19BCE1438**

**Course Code: CSE4001**

**Faculty: Dr. Harini S**

**Lab Experiment 7**

1. Sample - Barrier

Code:

#include <stdio.h>

#include <omp.h>

int main()

{

    int n = 7;

    int a[] = {1,7,3,9,2,4,12};

    int y[n], b[n];

    #pragma omp barrier

    for(int i = 1; i < n; i++)

    {

        b[i] = (a[i-1] + a[i]) / 2.0;

        printf("b[%d] = %d\n", i, b[i]);

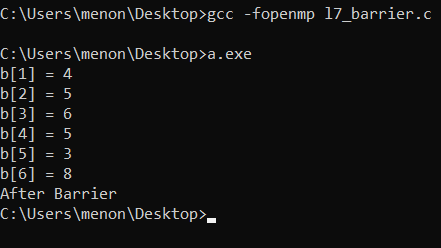
    }

    #pragma omp single

        printf("After Barrier");

}

Output:



1. Sample - nowait

Code:

#include<stdio.h>

#include<stdlib.h>

#include<omp.h>

int main()

{

    int a[] = {1, 2, 3, 4, 5};

    float b[5], y[5];

    int z[] = {64, 4, 196, 123, 224};

    #pragma omp parallel

    {

        #pragma omp for nowait

            for(int i = 1; i < 5; i++)

            {

                b[i] = (a[i] + a[i-1]) / 2.0;

                printf("b[%d] = %f\n", i, b[i]);

            }

        #pragma omp for nowait

            for(int i = 0; i < 5; i++)

            {

                y[i] = z[i]/2.0;

                printf("y[%d] = %f\n", i, y[i]);

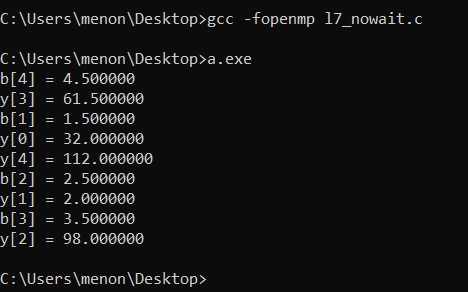
            }

    }

    return 0;

}

Output:



1. Sample – Parallel For (Nested)

Code:

#include <stdio.h>

#include <omp.h>

int main()

{

    int n = 3;

    int a[3][3] = {

        {1,2,3},

        {4,5,6},

        {7,8,9}

    };

    for(int i=1;i<n;i++)

    {

        #pragma omp parallel for

        for(int j=0;j<n;j++)

        {

            a[i][j] = a[i-1][j] + 2;

        }

    }

    for(int i=0;i<n;i++)

    {

        for(int j=0;j<n;j++)

        {

            printf("a[%d][%d] = %d\n", i, j, a[i][j]);

        }

    }

}

Output:

