Parallel Programming
LAB 2 -13th August 2018
Note: Write all programs in your observation book and record the results. Get the signature of faculty /teaching assistance.

Objective: To learn the various clauses of for construct:

1. Learn the working of lastprivate() clause:

```
#include<stdio.h>
#include<omp.h>
void main()
{
int x=0,i,n;
printf("Enter the value of n");
scanf("%d",&n);
#pragma omp parallel
{
        int id=omp_get_thread_num();
        #pragma omp for lastprivate(i)
        for(i=0;i<n;i++)
        {
        printf("Thread %d: value of i : %d\n",id,i);
        x=x+i;
        printf("Thread %d: x is %d\n",id,x);
        }
}
printf("x is %d\n",x);
printf("i IS %d\n",i);
}
```

2. Demonstration of nowait clause:
Check the output by removing nowait clause.

```
#include<stdio.h>
#include<omp.h>
void main()
{
int i,n;
double t1,t2;
printf("Enter the value of n");
scanf("%d",&n);
t1=omp_get_wtime();
#pragma omp parallel num_threads(4)
        {
        int id=omp_get_thread_num();
        #pragma omp for nowait
        for(i=0;i<n;i++)
        printf("Thread %d: value of i : %d\n",id,i);
        printf("\nI am Thread %d NO WAIT EFFECT\n",id);
        }
t2=omp_get_wtime();
printf("Time taken is %f",t2-t1);
}
```

3. Demonstration of Ordered Clause

```
#include<stdio.h>
#include<omp.h>
void main()
{
int i,n,a[50],b[50],sum;
double t1,t2;
printf("Enter the value of n");
scanf("%d",&n);
t1=omp_get_wtime();
#pragma omp parallel num_threads(4)
        {
        int id=omp_get_thread_num();
        #pragma omp for ordered reduction(+:sum)
        for(i=0;i<n;i++)
        {
        printf("Thread %d: value of i : %d\n",id,i);
        sum=sum+i;
                #pragma omp ordered
                {
                b[i]=i+1;
                printf("b[%d] value is %d in ORDER\n",i,b[i]);
                }
        }
        }
t2=omp_get_wtime();
printf("Time taken is %f",t2-t1);
}
```

4. Demonstration of collapse clause

```
#include<stdio.h>
#include<omp.h>
void main()
{
int i,j,n,a[50][50];
double t1,t2;
printf("Enter the value of n");
scanf("%d",&n);
t1=omp_get_wtime();
#pragma omp parallel num_threads(4)
        {
        int id=omp_get_thread_num();
        #pragma omp for collapse(2)
        for(i=0;i<n;i++)
          for(j=0;j<n;j++)
                {
                a[i][j]=i+j;
                printf("a[%d][%d] is %d\n",i,j,a[i][j]);
                }
        }
t2=omp_get_wtime();
printf("Time taken is %f",t2-t1);
}
```

5. Assignment:

Write a C/C++ parallel program to find sum of elements in an array.