Practical No. 3 (Batch B4)

Study and Implementation of schedule, nowait, reduction, ordered and collapse clauses

Q1: Analyse and implement a Parallel code for below program using OpenMP. (Demonstrate the use of different clauses wherever applicable)

```
/* Program to compute Pi using Monte Carlo methods */

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include <string.h>

#define SEED 35791246


main(int argc, char* argv)

{

  int niter=0;

  double x,y;

  int i,count=0; /* # of points in the 1st quadrant of unit circle */

  double z;

  double pi;


  printf("Enter the number of iterations used to estimate pi: ");

  scanf("%d",&niter);
```

```
    /* initialize random numbers */
    srand(SEED);
    count=0;
    for ( i=0; i<niter; i++) {
      x = (double)rand()/RAND_MAX;
      y = (double)rand()/RAND_MAX;
      z = x*x+y*y;
      if (z<=1) count++;
      }
    pi=(double)count/niter*4;
    printf("# of trials= %d , estimate of pi is %g \n",niter,pi);
  }
```

Q2: Write a code in OpenMP for multiplication of 2D Matrix and a 1D vector (Demonstrate the use of different clauses wherever applicable), vary the size of your matrices from 200, 1000, 2000 and 3000 and measure the runtime with one thread (Use functions in C in calculate the execution time or use GPROF)

For each matrix size, change the number of threads from 2,4,8., and plot the speedup versus the number of threads.

Explain whether or not the scaling behaviour is as expected.

Q3. For 2D Matrix Addition of size 2000, Write a OpenMP code with the following:

i. **Use STATIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup.**
ii. **Use DYNAMIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup.**
iii. **Demonstrate the use of nowait clause.**