Laboratorio: DESARROLLO DE APLICACIONES PARALELAS USANDO OPENMP.

Objetivos:

- Identificar las partes del código paralelizables.

- Proponer las directivas (pragmas) OpenMP de acuerdo al problema tratado.

-Usar funciones de la bibloteca OpenMP

PREPARACIÓN PREVIA: Estudie la conferencia , “Programación Paralela en sistemas de memoria compartida. OpenMP” y los ejercicios de la clase práctica

Temario

1- Detecte los errores en los programas omp_error_1.c, omp_error_2.c y omp_error_4.c que aparecen listados en la carpeta *Bugs.* Proponga las soluciones.

2- Implemente un programa paralelo utilizando OpenMP que calcule la media de los elementos de un arreglo. Para inicializar el arreglo puede emplear la siguiente función y puede utilizar la versión serie del cálculo de la media

```
void randInit(int array[], const int n, const int bound) {
    srand(time(NULL));
    for (int i = 0; i < n; i++)
        array[i] = rand() % bound;
}

double mean(int array[], const int n) {
    double sum = 0;
    for (int i = 0; i < n; i++)
        sum += array[i];
    return (double)sum / n;
}
```

A) Calcule el speedup para valores grandes de n y diferentes números de hilos
B) Modifique el tipo de distribución y el tamaño del fragmento y muestre el tiempo de ejecución en cada caso

3- El siguiente código secuencial implementa el producto de una matriz B de dimensión *N x N* por un vector *c* de dimension *N*

```
void prodmv(double a[N], double c[N], double B[N][N] )
{
  int i, j; double sum;
  for (i = 0; i < N; i++) {
    sum=0;
    for(j=0;j<N;j++)
        sum += B[i][j]*c[j];
    a[i]=sum;
  }
}
```

A) Escriba un programa paralelo con OpenMP del código dado
B) Calcule el speedup y la eficiencia

4- Implemente un programa paralelo con OpenMP que permita determinar el menor y mayor elemento de una matriz

5- Desarrolle un programa paralelo que permita determinar la cantidad de números primos en un rango de [1..N]. Puede basarse en la siguiente función

```
/******************************************************************************/

int prime_number ( int n )

/******************************************************************************/
/*
  Objetivo:

    retorna la cantidad de primos entre 1 y N.

  Parámetros:

    Input, int N, valor máximo a chequear.

    Output, int PRIME_NUMBER, cantidad de primos
*/
{
  int i;
  int j;
  int prime;
  int total = 0;

  for ( i = 2; i <= n; i++ )
  {
    prime = 1;

    for ( j = 2; j < i; j++ )
    {
      if ( i % j == 0 )
      {
        prime = 0;
        break;
      }
    }
    total = total + prime;
  }

  return total;
}
```