IN3200/IN4200 Exercise Set 7

Exercise 1

For each of the following code segments, use OpenMP pragmas to make the loop parallel, or explain why the code segmenet is not suitable for parallel execution.

a.

```
for (i=0; i < (int) sqrt(x); i++) {
  a[i] = 2.3 * x;
  if (i < 10) b[i] = a[i];
}
```

b.

```
flag = 0;
for (i = 0; (i<n) & (!flag); i++) {
  a[i] = 2.3 * i;
  if (a[i] < b[i]) flag = 1;
}
```

c.

```
for (i = 0; i < n; i++)
  a[i] = foo(i);
```

d.

```
for (i = 0; i < n; i++) {
  a[i] = foo(i);
  if (a[i] < b[i]) a[i] = b[i];
}
```

e.

```
for (i = 0; i < n; i++) {
  a[i] = foo(i);
  if (a[i] < b[i]) break;
}
```

f.

```
dotp = 0;
for (i = 0; i < n; i++)
  dotp += a[i] * b[i];
```

g.

```
for (i = k; i < 2*k; i++)
  a[i] = a[i] + a[i-k];
```

h.

```
for (i = k; i < n; i++)
  a[i] = b * a[i-k];
```

Exercise 2

Write a simple C code to compute the dot-product of two very long vectors. Use `#pragma omp parallel for` to do the parallelization. Choose different schedulers and chunksizes and observe the time usage.

```
dotp = 0.;
for (int i = 0; i < n; i++)
  dotp += a[i]*b[i];
```

Exercise 3

The following function can be used to compute a dense matrix-vector multiplication:

$$\mathbf{x} = \mathbf{A}\mathbf{y}$$

where $\mathbf{x}$ is a vector of length m, $\mathbf{A}$ is a dense $m \times n$ matrix, and $\mathbf{y}$ is a vector of length n.

```
void dense_mat_vec(int m, int n, double *x, double *A, double *y)
{
  int i, j;
  for (i=0; i<m; i++)
  {
    double tmp = 0.;
    for (j=0; j<n; j++)
      tmp += A[i*n+j]*y[j];
    x[i] = tmp;
  }
}
```

a.

Write a complete serial C program that

- accepts the values of m and n from command line at runtime;
- allocates the matrix $\mathbf{A}$, and the vectors $\mathbf{x}$ and $\mathbf{y}$;
- initializes $\mathbf{A}$ and $\mathbf{y}$ with some suitable values;
- calls the function `dense_mat_vec`.

b.

Insert appropriate OpenMP pragma(s) in the function `dense_mat_vec` (and possibly also in the serial function `main`) to create a parallel program.

c.

If you are allowed to insert OpenMP pragmas in the function `main` but not in the function `dense_mat_vec`, how would you create an OpenMP parallel implementation?