# Exercise 2

**Task 2.1** (OpenMP theoretical concepts)**.** Recall what you have learned in the classes.

a) Indicate and describe the types of scheduling that OpenMP provides. Indicate the situations for which they are indicated.

b) What are the two main disadvantages of dynamic scheduling?

c) What forms of synchronization are provided by OpenMP? Specify and fully describe each primitive/construct/clause.

**Task 2.2.** Complete the scoping of the variables in the code using their types accordingly.

```
#pragma omp parallel for
for (i=0; i<m; i++){
     y[i]=0;
     for (j=0; j<n; j++){
          y[i]= y[i] + A[i][j]*x[j];
     }
}
```

**Task 2.3.** Eliminate any data race in the code. *fib(i)* is a function that outputs the $i^{th}$ element in the Fibonacci sequence.

```
int y[55]={-1};
#pragma omp parallel for
for (i=0; i<10; i++){
     y[fib(i)]=fib(i)+i;
}
```

**Task 2.4.** Indicate the type of dependence (if any) in each piece of code and parallelize them but removing the dependence accordingly.

a)
```
#pragma omp parallel for
for (i=0; i<n-1; i++){
     y[i]=y[i+1]*x[i+1];
}
```

b)
```
#pragma omp parallel for
for (i=0; i<n; i++){
                y[i] = h[f(1)] * y[i];
                d = y[i];
}
printf("Result = %d.\n",d);
```

c)
```
#pragma omp parallel for
for (i=4; i<n-2; i+=2){
                a[i+1] = x;
                a[i+2] = y;
                h[i] = a[i];
                h[i+1] = a[i];
}
```

Discussion of this exercise on $13^{th}$ November 2014.