

Utilitzant el model vist en el problema anterior, resolgueu el següent problema d'examen:
Problema: Es pretén paral·lelitzar el següent codi en C mitjançant OpenMP:

```
float dist(int n, float *d1, float *d2) {
    float sum=0.0;
    float x1,x2;
    for(int i=0;i<n;i++) {
        x1=d1[i];
        for(int j=i+1;j<n;j++) {
            x2=d2[j];
            sum+=pow((x1-x2),2);
        }
    }
    return sqrt(sum)/(n*n);
}
```

(+,-,*) tendran un coste de 1 FLOP

(exp, sqrt) tendran un coste de 3 FLOP

Tenemos dos bucles for.

```
for(int i=0;i<n;i++) {      ----> n iteraciones      -----
    x1=d1[i];              -----
    for(int j=i+1; j<n; j++){ --> n-i+1 iteraciones -----
        x2=d2[j];          -----
        sum += pow((x1-x2),2); --> 3+1+1 FLOP -----
    }
}
return sqrt(sum)/(n*n)    3+1+1
```

$10(n^2 + n(1-i)) + 5 \text{ FLOPS}$

Paralelización.

x1=d1[i] no tiene riesgos.

x2=d2[j] no tiene riesgos.

sum += pow((x1-x2),2); depende de las dos operaciones anterior por lo que debe de estar detrás.
 De las dos asignaciones.

for(int j=i+1; j<n; j++) debe de estar detrás del bucle de i.

Propuesta de paralelización con openmp

```
#pragma omp parallel for private(sum, i, j) shared(d1*, d2*) reduction(+:sum)
{
    for(int i=0;i<n;i++) {
        for(int j=i+1; j<n; j++){
            sum += pow((d1[i]-d2[j]),2);
        }
    }
    return sqrt(sum)/(n*n)    3+1+1
}
```