1. Indica cuál será el valor de la variable n al final de la ejecución del siguiente código:

```
int n =0;
#pragma omp parallel for reduction(*:n)
for (int i =n; i< size; ++i)
      n *= i;
```

   a. Dependerá del valor de la variable *size*.
   b. 0 ←
   c. El valor de *n* será igual a *size* - 1
   d. Ninguna respuesta es correcta.

2. Sobre el código que aparece a continuación, ¿qué afirmación es correcta?

```
#pragma omp parallel private(sumalocal)
{
      sumalocal =0;
      #pragma omp  for
      for(i=0; i<n; i++)
            sumalocal += a[i];
      #pragma omp barrier
      #pragma omp critial
            suma = suma + sumalocal;
      #pragma omp barrier
      #pragma omp critial
            printf("La suma es =%d\n", suma);
}
```

   a. El valor de suma que se imprime sería correcto si cambiamos *private(sumalocal)* por *private(sumalocal, suma)* No
   b. El valor de suma que se imprime no es siempre correcto No
   c. Todas las demás respuestas son incorrectas No, la d es cierta
   d. Uno de los *#pragma omp barrier* es innecesario Cierto, el primero

3. ¿Cuál es la única directiva con la que se puede usar la cláusula *copyprivate*?
   a. Se puede usar con más de una directiva
   b. atomic
   c. master
   d. single ←

4. Indica qué valor tendrá la variable ret después de ejecutar la siguiente reducción cuando se ejecuta con 4 hebras:

```
int i, n=6, ret = 1;
#pragma omp parallel reduction(+:ret)
for(i=omp_get_thread_num(); i<omp_get_max_threads();i+=omp_get_num_threads())
      ret +=i;
```

   a. El valor de ret será indeterminado porque existe una condición de carrera → Debido a i
   b. 5
   c. 7
   d. 3

5. ¿Cuál de las siguientes afirmaciones es correcta?

a. Ninguna otra respuesta es correcta
b. Las directivas ajustan el comportamiento de las cláusulas
c. Las cláusulas ajustan el comportamiento de las directivas
d. Directivas y cláusulas son iguales

6. ¿Cuáles de las siguientes cláusulas crean instancias privadas de una variable con un valor indefinido?
   a. *private* y *firstprivate*
   b. *copyprivate* y *firstprivate*
   c. *private* y *lastprivate*
   d. *firstprivate* y *lastprivate*

7. ¿Cuál de los siguientes fragmentos de código tardará menos en calcular la sumatoria de los primeros N=225 números impares en una máquina con 4 procesadores?
   a.
```
long sum =0;
#pragma omp parallel sections
{
  #pragma omp section
  for (long i =1; i<N; i+= 2)
       #pragma omp atomic
            sum += i;
}
```
   b.
```
long sum =0;
#pragma omp parallel
{
  long p =0;
  #pragma omp for
  for (long i =1; i<N; i+= 2)
       p += i;
  #pragma omp atomic
       sum += p;
}
```
   c.
```
long sum =0;
#pragma omp parallel for
for (long i =1; i<N; i+= 2)
  #pragma omp atomic
       sum += i;
```
   d.
```
long sum =0;
#pragma omp parallel sections
{
  #pragma omp section
  for (long i =1; i<N; i+= 4)
       #pragma omp atomic
            sum += i;
  #pragma omp section
  for (long i =3; i<N; i += 4)
       #pragma omp atomic
            cum += i;
```

```
}
```

8. ¿Cuánto vale *n* al final?

```
int n =1;
#pragma omp parallel for reduction(*:n)
for (int i =n; i< 5; ++i)
      n *= i;
return n;
```

a. 0
b. 1
c. 24 ←
d. 6

9. ¿Existe algún problema que impida compilar el siguiente código?

```
int n =1;
#pragma omp parallel for default(none)
for (int i = 0; i< 5; ++i)
      n += i;
```

a. El ámbito de la variable *n* no está definido. ←
b. Ninguna otra respuesta es correcta
c. El ámbito de la variable *i* no está definido
d. Existe una condición de carrera

10. Sobre el código que aparece a continuación, ¿qué afirmación es correcta?

```
#pragma omp parallel private(sumalocal)
{
      sumalocal =0;
      #pragma omp  for
      for(i=0; i<n; i++)
            sumalocal += a[i];
      #pragma omp barrier
      #pragma omp critical
            suma = suma + sumalocal;
      #pragma omp barrier
      #pragma omp single
            printf("La suma es =%d\n", suma);
}
```

a. El valor de suma que se imprime es correcto Cierto
b. Tendríamos el mismo comportamiento si cambiamos *critical* por *atomic* Cierto, lo he probado
c. Tendríamos el mismo comportamiento si eliminamos los dos *#pragma omp barrier* No, el segundo barrier es necesario
d. Todas las demás respuestas son incorrectas No, a y b son ciertas