

8. Analiza el siguiente código e indica qué nos dirá el compilador referido a las siguientes líneas de código

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

- a) La variable i no puede ser privada
- b) La variable n no está especificada
- c) La variable n no está especificada como privada
- d) Ninguna opción de las anteriores

9. Indica qué valor tendrá la variable ret después de ejecutar la siguiente reducción cuando se ejecuta con 4 hebras \rightarrow Sus id son 0,1,2,3, luego $1+1+2+3=7$

```
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) 5
- b) 7
- c) El valor de ret será indeterminado porque existe una condición de carrera ret es privada
- d) 3

10. Asumiendo que v2 de dimensión N y que todos sus elementos están inicializados a cero, ¿cuál de los siguientes códigos calcula de forma correcta el producto de la matriz m (dimensión NxN) por el vector v1 (dimensión N) paralelizando el bucle que recorre las columnas?

- a)

```
#pragma omp parallel private(j) for(i=0;i<N;i++){ #pragma omp for
reduction(+:v2[i]) for(j=0;j<N;j++){ v2[i] += m[i][j]*v1[j];} }
```
- b)

```
#pragma omp parallel for private(j) for(i=0;i<N;i++){ for(j=0;j<N;j++){ v2[i]
+= m[i][j]*v1[j];} }
```
- c)

```
#pragma omp parallel private(i,j) for(i=0;i<N;i++){ for(j=0;j<N;j++){ v2[i] +=
m[i][j]*v1[j];} }
```
- d)

```
#pragma omp parallel private(i) for(i=0;i<N;i++){ #pragma omp for reduction(+:v2[i])
for(j=0;j<N;j++){ v2[i] += m[i][j]*v1[j];} }
```

11. ¿Cuál de las siguientes directivas no admite ninguna cláusula?

- a) single
- b) parallel
- c) Todas las directivas admiten alguna cláusula
- d) critical

12. Supongamos una máquina en la que el número de hebras de las que se puede disponer para ejecutar zonas paralelas de código es limitado. En esas condiciones ¿qué valdrá la variable n al terminar de ejecutar el siguiente código?

```
int n = 1;
#pragma omp parallel for firstprivate(n) lastprivate(n)
    for (int i = 0; i < 5; ++i)
        n += 1;
```

- a) n al salir del bucle está indefinido porque es privada
- b) Dependerá del número de hebras que lo ejecuten en cada momento**
- c) 1
- d) 5

12. ¿Cuál de las siguientes directivas no admite ninguna cláusula?

- a) single
- b) critical**
- c) parallel
- d) Todas las directivas admiten alguna cláusula

13. Supongamos una máquina en la que el número de hebras de las que se puede disponer para ejecutar zonas paralelas de código es limitado. En esas condiciones, ¿qué valdrá la variable n al terminar de ejecutar el siguiente código?

```
int n = 1;
#pragma omp parallel for firstprivate(n) lastprivate(n)
    for (int i = 0; i < 5; ++i)
        n += i;
```

- a) n al salir del bucle está indefinida porque es privada
- b) Dependerá del número de hebras que lo ejecuten en cada momento**
- c) 1
- d) 5

14. ¿Cuál será el valor de n tras ejecutar el siguiente código?

```
int i, n=2;
#pragma omp parallel shared(n) private(i)
    for(i=0; i < 4; i++){
        #pragma omp single
        {
            n += i;
        }
    }
```

- a) 16
- b) 8**
- c) 2
- d) Indeterminado

15. ¿Cuál de los siguientes fragmentos de código paralelo calcula correctamente la sumatoria de los primeros números impares hasta el $N = 1000$?

- a) `int sum = 0;`
`#pragma omp parallel for`
`for(long i = 1; i < N; i += 2)`
`sum += i;` *Sum es compartida, habrá hebras que escriban simult. en sum*
- b) `int sum = 0;`
`#pragma omp parallel`
`for(long i = 1; i < N; i += 2)`
`sum += i;` *Cada hebra ejecutará el bucle entero y a la vez que las demás*
- c) `int sum = 0;`
`#pragma omp parallel sections{`
`#pragma omp section`
`for (long i = 1; i < N; i += 4)`
`sum += i;`
`#pragma omp section`
`for (long i = 3; i < N; i += 4)`
`sum += i;`
`}` *2 hebras suman la mitad del total de los impares lo hacen a la vez*

d) Ninguna otra respuesta es correcta

16. Indica cual 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;`

$(1 + i + i + 1 + \dots + \text{size} - 1)^{\text{n}^{\circ} \text{hebras}}$

- a) Ninguna respuesta es correcta
b) El valor de n será igual a $\text{size} - 1$
c) 0

d) Dependerá del valor de la variable size

17. ¿Cuánto vale ret al final?

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

Si hubiese 3 hebras, le tocan 2 iter. a cada una:

*Hebra 0 {1+0+3}=4
 " " 1 {1+1+4}=6
 " " 2 {1+2+5}=8 } $\oplus \rightarrow 18$*

- a) 15
b) 1
c) 16

d) Ninguna de las otras respuestas es correcta

18. ¿Cuál de las siguientes afirmaciones es correcta?

a) Las cláusulas ajustan el comportamiento de las directivas

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