



# Examen BP2



Universidad de Granada - Doble Grado en Ingeniería Informática y Matemáticas  
Arquitectura de Computadores



Desconocido: 45338112 Guerrero Cano, Valentín



Inicio: Hoy, miércoles, 09:40:07

Final: Hoy, miércoles, 09:54:48

Preguntas: 10

Respuestas  
válidas:

Puntuación:

Nota:

1

¿Cuánto vale `sum` al final si `OMP_NUM_THREADS = 3`?

Elección única

```
int i = 0, sum = 0;
#pragma omp parallel shared(i)
for (i = 0; i < 5; ++i)
    #pragma omp atomic
    sum += i;
return sum;
```

Usuario Profesores

- a) Indeterminado, existe condición de carrera
- b) 30
- c) 5
- d) 10

2

Elección única

Si se quiere difundir el valor de una variable inicializada por una hebra en una directiva `single` a las variables del mismo nombre del resto de hebras. ¿Cuál sería la cláusula que se tendría que utilizar?

Usuario Profesores

- a) `firstprivate`
- b) Ninguna de las anteriores
- c) `lastprivate`
- d) `copyprivate`

3

Observando el siguiente código. ¿Cuál es el error o los errores para que el programa compile correctamente y obtenga el resultado deseado?

## Elección única

```
int n = 1, constant;
#pragma omp parallel for reduction(*n) private(constant)
for (int i = 0; i < omp_get_max_threads(); ++i)
    n *= (i + constant);
return n;
```

Usuario Profesores

- ☐ a) Todas las otras respuestas son correctas
- ☐ b) Hay que cambiar la cláusula `private` por `firstprivate`
- ☐ c) La cláusula `reduction` tiene una sintaxis errónea.
- ☐ d) La variable `constant` debe ser inicializada

4

## Elección única

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

Usuario Profesores

- ☐ a) 

```
int sum = 0;
#pragma omp parallel
for (long i = 1; i < N; i += 2)
    sum += i;
```
- ☐ b) 

```
int sum = 0;
#pragma omp parallel for
for (long i = 1; i < N; i += 2)
    sum += i;
```
- ☐ c) Ninguna otra respuesta es correcta
- ☐ d) 

```
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;
}
```

5

## Elección única

¿Cuál es la única directiva con la que se puede usar la cláusula `copyprivate`?

Usuario Profesores

- ☐ a) `master`
- ☐ b) Se puede usar con más de una directiva
- ☐ c) `single`
- ☐ d) `atomic`

6

## Elección única

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);
}

```

Usuario Profesores

- ☐ a) Tendríamos el mismo comportamiento si eliminamos los dos `#pragma omp barrier`
- ☐ b) Tendríamos el mismo comportamiento si cambiamos `critical` por `atomic`
- ☐ c) Todas las demás respuestas son incorrectas
- ☒ d) El valor de suma que se imprime es correcto

7

Elección única

Asumiendo que `v2` es 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?

Usuario Profesores

- ☐ a) 

```
#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];
    }
}
```
- ☐ b) 

```
#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];
    }
}
```
- ☐ c) 

```
#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];
    }
}
```
- ☐ d) 

```
#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];
    }
}
```

8

Elección única

¿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;
    }
}

```

Usuario Profesores

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

**9**

Elección única

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;
```

Usuario Profesores

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

**10**

Elección única

¿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;
```

Usuario Profesores

- ☐ a) 0
- ☐ b) 24
- ☐ c) 6
- ☐ d) 1