

# testBP2preguntas.pdf



**Cristinasj**



**Arquitectura de Computadores**



**2º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**

**test BP2**  
**PREGUNTAS**  
**(Ver tambien el PDF de respuestas)**

1. V o F. Las directivas que aceptan cláusulas son parallel, for, sections, single y workshare
2. V o F. Las directivas de control de ámbito de las variables son shared, private, lastprivate, firstprivate, default y copyprivate
3. V o F. Por lo general, las variables declaradas dentro de una región son privadas
4. V o F. La cláusula shared solo sirve para arrays
5. V o F. OMP se ocupa de la exclusión mutua. Por lo tanto, podemos leer y escribir en una variable modificada por la cláusula shared sin problema.
6. V o F. Los dos siguientes fragmentos de código son equivalentes

1) #pragma omp parallel for  
for (i=0;i<n;i++)  
    a[i] += i;

2) #pragma omp parallel for **shared(a)**  
for (i=0;i<n;i++)  
    a[i] += i;

7. V o F. Para el siguiente código, daría error si olvidamos compilar con -fopenmp

```
#include <stdio.h>
#ifdef _OPENMP
    #include <omp.h>
#endif

main()
{
    int i, n = 7;
    int a[n];

    for (i=0; i<n; i++)
        a[i] = i+1;

    #pragma omp parallel for shared(a)
    for (i=0; i<n; i++)    a[i] += i;

    printf("Después de parallel for:\n");

    for (i=0; i<n; i++)
        printf("a[%d] = %d\n",i,a[i]);
}
```

8.V o F. La ejecución del siguiente código es:

X

X

X

X

```
OMP_NUM_THREADS=2
```

```
a='x';
```

```
puts(a);
```

```
#pragma omp prarallel private(a)
```

```
{puts(a)}
```

```
puts(a);
```

9. V o F. A partir del siguiente código

```
OMP_NUM_THREADS=2  
a='x';  
puts(a);  
#pragma omp prarallel private(a)  
    {puts(a)}  
puts(a);
```

Podemos estar seguros de que al menos la segunda letra en salir es x

10. V o F. Se puede hacer lastprivate con una directiva parallel

11. Cual es la salida por pantalla

```
code
#include <stdio.h>
#include <omp.h>

int main () {
    int a = 3;
    #pragma omp parallel for lastprivate(a)
    { a = 4; }
    printf("%d",a);
    return 0;
}
```

12. Escribir el resultado por pantalla de lo siguiente

```
code
#include <stdio.h>
#include <omp.h>

int main () {
    int a = 3;
    #pragma omp parallel for lastprivate(a)
    for (int i=0; i<1; i++)
    { printf("%d",a); }
    return 0;
}
```

13. V o F. Siempre que se use default, el programador tienen que especificar el alcance de todas las variables que se vayan a utilizar

14. V o F. La directiva copyprivate solo se puede usar con single

15. V o F. Utilizar copyprivate es útil cuando se quiere hacer una lectura de un valor por teclado y llenar un vector con ella

16. V o F. Se puede hacer dos bucles for anidados con parallel, de la siguiente forma

```
}  
#pragma omp for  
for (int i=0; i<tam*tam; i++)  
{  
    #pragma omp for  
    for (int j=0; j<tam; j++)  
        v1[i][j]=i*j*0.1;  
}  
//
```

17. Escribir la función de OMP que calcula el tiempo de un trozo de código