

Nombre: ..... DNI: .....

Convocatoria: **Septiembre** Semana: **Nacional / Unión Europea RESERVA** Tipo de Examen: **B**

- El **test** debe ser contestado en la **hoja de lectura óptica**. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta.
- El test es eliminatorio y aporta un 30% de la nota final. Son necesarias 8 respuestas correctas (6 con las prácticas aprobadas) para que se corrija el ejercicio.
- Cada respuesta correcta: 1 punto. Respuesta incorrecta o en blanco: 0 puntos.

1.- ¿Cuál de los siguientes objetivos de la programación es el más prioritario?

- a) Claridad
- b) Corrección
- c) Eficiencia
- d) Portabilidad

2.- El siguiente fragmento de código en C± muestra por pantalla:

```
int x = 5; int z = 2; int y;  
x = z * 4;  
y = x % z;  
printf("%d%.2f%d", x, float(y), z);
```

- a) 8.2
- b) 8.002
- c) 80.002
- d) 800.02

3.- En la declaración correcta en C± del procedimiento:

```
void Procedimiento( TipoAlgo MiArgumento ) {...}
```

- a) MiArgumento es un argumento por valor o por referencia según el tipo TipoAlgo
- b) MiArgumento siempre es un argumento por valor
- c) MiArgumento siempre es un argumento por referencia
- d) MiArgumento es el tipo del argumento TipoAlgo

4.- En la expresión aritmética  $a - 4.2/b * c + d$ , la tercera operación realizada es:

- a) La división
- b) La multiplicación
- c) La suma
- d) La resta

5.- La doble referencia se produce:

- a) Cuando un mismo elemento se utiliza con dos nombres distintos
- b) Cuando dos punteros apuntan al mismo tipo de elemento
- c) Cuando se produce una redefinición de elementos
- d) Cuando dos índices indexan el mismo elemento de una formación

6.- La programación a la defensiva:

- a) Aumenta la eficiencia del programa
- b) Facilita el desarrollo ascendente
- c) Corrige las situaciones de error
- d) Produce programas robustos

7.-Cuál de las siguientes estructuras NO es equivalente a:

```
do { Sentencias } while (Condicion);
```

- a) while (Condicion) { Sentencias }
- b) Sentencias; while (Condicion) { Sentencias }
- c) seguir=true; while (seguir) { Sentencias; seguir=Condicion; }
- d) seguir=true; while (seguir || Condicion) { Sentencias; seguir=false; }

8.- En C±, la sentencia struct:

- a) Permite definir un esquema unión
- b) Sólo permite definir registros
- c) Sólo permite definir tipos abstractos de datos
- d) Permite definir un registro o un tipo abstracto de datos

9.- Supongamos el fragmento de código en C±:

```
.....  
void Func_1(int & a, int & b) {  
    a = 4;  
    b = b*a;  
}  
.....  
b = 5;  
Func_1(b,b);
```

Después de la ejecución de este fragmento:

- a) b vale 20
- b) b vale 16
- c) b vale 5
- d) b vale 9

10.- En C±, el uso de la sentencia break:

- a) Es opcional en los casos de un switch
- b) Es obligatoria en todos los casos de un switch
- c) Se puede utilizar en cualquier punto de un programa
- d) Es obligatoria en la selección por defecto

## EJERCICIO DE PROGRAMACIÓN

Realizar en C± un tipo abstracto de datos (TAD) para gestionar un conjunto de pisos en alquiler de hasta 10 pisos. La información que se dispone de cada piso es el nombre de la persona que lo pone en alquiler, el teléfono, el precio y el tamaño (pequeño, mediano, grande). Las operaciones a realizar son: 1.- Insertar un nuevo piso en alquiler. 2.- Devolver el subconjunto de pisos cuyo precio sea inferior a un precio dado. 3.- Imprimir por pantalla los datos: nombre de la persona que lo pone en alquiler, teléfono, precio y tamaño, de todos los pisos en alquiler existentes.