FUNDAMENTOS DE PROGRAMACION	MODELO 2	*CONV. ORD. 24/25*
GRADO DE INFORMÁTICA y GRADO DE TECNOLOGÍAS DE LA INFORMACIÓN-ETSII-UNED		
MATERIAL AUTORIZADO: **NINGUNO**		DURACIÓN: **2 HORAS**

INSTRUCCIONES:	**1) El test debe ser contestado en la hoja de marcas. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta.** **2) El test es eliminatorio y aporta un 30% de la nota final. Son necesarias 8 preguntas correctas (6 con las prácticas aprobadas) para que se corrija el ejercicio.** **3) Cada respuesta correcta 1 pto. Respuesta incorrecta, doble o en blanco: 0 pto.** **4) Puede quedarse, si lo desea, con esta hoja de examen.**

CUESTIONES DE TEST

1. En el fragmento de programa C±:

```
case Uno : Dos; break;
```

A. Dos es una sentencia
B. Uno es una variable
C. Uno es una sentencia
D. Dos es una constante

2. Según el Manual de estilo de C± sobre la separación de la parte declarativa y ejecutiva de un bloque:

A. Es posible mezclar
B. Es obligatoria la separación
C. Es recomendable mezclar
D. Es recomendable separar

3. La ejecución del siguiente código:

```
for (int i = 1; i < LongitudVector; i++) {
  aux = miVector[i];
  j = i;
  while ( (j > 0) && (aux > miVector[j-1]) ) {
    miVector[j] = miVector[j-1];
    j--;
  }
  miVector[j] = aux;
}
```

A. Ordena miVector de menor a mayor
B. No termina nunca
C. Ordena miVector de mayor a menor
D. Va almacenando en aux el mayor elemento de miVector

4. La sentencia C±:

```
while (Uno) { Dos }
```

A. Dos es una expresión
B. Uno es una declaración
C. Uno es una expresión
D. Dos es una declaración

5. Para declarar un registro con variantes ….

A. sólo se necesita utilizar union
B. sólo se necesita utilizar struct
C. se necesita utilizar siempre struct y union
D. sólo se necesita utilizar struct o bien unión

6. La ejecución del siguiente código:

```
#include "stdio.h"

int Global;
int Ext= K;

void Inside (){ const int Ext=K;   Global = Ext*Ext; }

void Inside_Sec() { Inside();  Ext=Global+ K; }

void Prime(int & Ext, int & Inside){
  Inside_Sec();   Inside = Ext -  K; }

int main(){
   int Inside =  K;
   Prime(Inside, Inside);
   printf("VALORES = %d - %d - %d\n",Global, Inside, Ext); }
```

¿Cuál sería el valor K para que se muestre el mensaje: "VALORES = 16 - 0 - 20"?

A. 0
B. 2
C. 4
D. 6

7. Elija la ordenación correcta, de menor a mayor, de las funciones O de crecimiento asintótico:

A. $O(n) < O(\log n) < O(n^k) < O(2^n)$
B. $O(\log n) < O(n) < O(n^k) < O(2^n)$
C. $O(\log n) < O(n) < O(2^n) < O(n^k)$
D. $O(n) < O(\log n) < O(2^n) < O(n^k)$

8. Dada la declaración en C±:

```
const int perimetro = 2*(largo+ancho);
```

Podemos afirmar que:

A. perimetro es una variable y largo una constante
B. perimetro y largo son variables
C. perimetro y largo son constantes
D. perimetro y ancho son variables

9. El fragmento de programa en C±:

```
void Uno::Dos ( ) { ...}
```

A. Es la interfaz del tipo abstracto Uno
B. Es la interfaz del dato encapsulado Uno
C. Es la implementación del procedimiento Dos de un dato encapsulado
D. Es la implementación del procedimiento Dos de un tipo abstracto

10. La programación estructurada recomienda el uso:

A. de compiladores
B. de intérpretes
C. de refinamientos sucesivos
D. del modelo abstracto de cómputo lógico

EJERCICIO DE PROGRAMACIÓN

Un colegio necesita un sistema para gestionar los horarios de las clases de hasta 15 asignaturas diferentes para 20 profesores. Cada clase tiene la siguiente información:

- Nombre de la asignatura (hasta 50 caracteres).
- ID del profesor (identificador numérico del profesor).
- Día de la semana (lunes, martes, miércoles, jueves, viernes).
- Hora de inicio (en formato 24 horas: cuatro dígitos, por ejemplo 6:00PM se escribe 1800, o 9:15AM se escribe 0915).
- Duración (en minutos).

Implemente un Tipo Abstracto de Datos (TAD) para gestionar los horarios de clases. El TAD dispondrá de las siguientes operaciones:

a) AgregarClase: Permite agregar una nueva clase su asignatura, el profesor, el día, la hora de inicio y la duración.
b) ListarHorarioProfesor: Dado el ID de un profesor, imprime todas las clases que tiene asignadas, organizadas por día y hora.
c) ImprimirTodosLosHorarios: Imprime los horarios de todos los profesores, mostrando las clases organizadas por día y hora.