

 71901020		Fundamentos de Programación			901	
		MULTIPLES GRADOS				71
		Febrero - 2026 Original	Duración: 120 min.	EXAMEN: Tipo B Mixto	Nacional - U.E.- GUINEA 1º Cuatrimestre	
Material: Ninguno					Hoja 1 de 3	

FUNDAMENTOS DE PROGRAMACION		MODELO 2	CONV. ORD. 25/26
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. ¿Cuál es la diferencia fundamental entre una función y un procedimiento en C±?

- A. Un procedimiento siempre devuelve un valor
- B. Los procedimientos siempre usan el paso de argumentos por valor
- C. Las funciones siempre usan el paso de argumentos por referencia
- D. Una función siempre devuelve un valor

2. En la notación Backus-Naur Form (BNF), los metasímbolos de repetición, opción y agrupación son respectivamente:

- A. [], {}, y ()
- B. {}, () y []
- C. (), [] y {}
- D. {}, [] y ()

3. Dado el código C± con una sentencia "for" anidada en un "while" y la variable f inicializada a 1:

```
while (f <= 3) {
    for (int c = 1; c <= 4; c++) {
        printf("Par: %d,%d\n", f, c);
    }
    f++;
}
```

¿Cuál sería el código equivalente usando dos sentencias "while"?

A.

```
while (f <= 3) {
    c = 1;
    while (c >= 4) {
        printf("Par: %d,%d\n", f, c);
        c--;
    }
    f++;
}
```

C.

```
while (f <= 3) {
    c = 1;
    while (c <= 4) {
        printf("Par: %d,%d\n", f, c);
        c++;
    }
    f++;
}
```

B.

```
while (f < 3) {
    c = 1;
    while (c < 4) {
        printf("Par: %d,%d\n", f, c);
        c++;
    }
    f++;
}
```

D.

```
while (f < 3) {
    c = 1;
    while (c < 4) {
        printf("Par: %d,%d\n", f, c);
        c--;
    }
    f--;
}
```

 71901020		Fundamentos de Programación			901	
		MULTIPLES GRADOS				71
		Febrero - 2026 Original	Duración: 120 min.	EXAMEN: Tipo B Mixto	Nacional - U.E.- GUINEA 1º Cuatrimestre	
Material: Ninguno					Hoja 2 de 3	

4. En la programación imperativa, las acciones primitivas son las:

- A. instrucciones más antiguas del procesador
- B. que realiza directamente el procesador
- C. que realiza indirectamente el procesador
- D. acciones más complejas que puede realizar el procesador

5. ¿Cuál es el propósito de la ocultación en un TAD?

- A. Hacer que el programa funcione más rápido
- B. Reducir el consumo de memoria
- C. Esconder nuestra identidad
- D. Permitir cambios en su implementación interna sin que eso afecte al código que usa el TAD

6. Respecto al comportamiento asintótico, si un programa A tiene complejidad logarítmica, el programa B tiene complejidad polinómica y el programa C tiene complejidad cuadrática entonces podemos afirmar que:

- A. el programa B es más eficiente que A y C
- B. el programa C es más eficiente que A y B
- C. el programa B tiene similar eficiencia que A y peor que C
- D. el programa A es más eficiente que B y C

7. ¿Qué imprime el siguiente código?

```
int x = 2;
int y = 10;
while (x < y) {
    switch (y % 3) {
        case 0: y = y + 1; break;
        default: x = x + 2;
    }
    printf("%d %d ", y, x);
}
```

- A. 10 4 10 6 10 8 10 10
- B. 10 4 10 6 10 8
- C. 4 10 6 10 8 10 10 10
- D. 4 10 6 10 8 10

8. El concepto de bucle corresponde al esquema básico de la estructura:

- A. de selección
- B. de iteración
- C. secuencial
- D. selección múltiple

9. En C±, para definir un tipo sinónimo es necesario usar:

- A. la palabra reservada array y un tipo ya definido
- B. un tipo ya definido y un módulo de definición
- C. la palabra reservada typedef y un tipo ya definido
- D. un tipo ya definido, un módulo de definición y un módulo de implementación

 71901020		Fundamentos de Programación			901	
		MULTIPLES GRADOS				71
		Febrero - 2026 Original	Duración: 120 min.	EXAMEN: Tipo B Mixto	Nacional - U.E.- GUINEA 1º Cuatrimestre	
Material: Ninguno					Hoja 3 de 3	

10. Dado el código en C±:

```
#include <stdio.h>

int recursivo(int P, int Q, int T) {
    if (P == 0) { return 1; }
    else {
        return P + T + recursivo(Q-1, T, P);
    }
}

int main() {
    int prim = K; int segun = L; int terce = L;
    int resultado = recursivo(prim, segun, terce);
}
```

¿Cuáles serían los valores de K y L para que el valor de la variable resultado sea 4?

- A. K=2 y L=1
- B. K=1 y L=2
- C. K=2 y L=2
- D. K=1 y L=0

EJERCICIO DE PROGRAMACIÓN

Realizar un tipo abstracto de datos (TAD) para gestionar un almacén de pinturas para exteriores. El almacén dispone de 10 depósitos de 1000 litros en los que se puede almacenar un tipo de pintura para exterior de los cinco tipos gestionados: pintura acrílica, pintura de resina de silicona, pintura de siloxano, pintura elástica o pintura de silicatos. Cada depósito tiene asociado un valor del precio de la pintura por litro.

El TAD dispondrá de los siguientes subprogramas:

- **CargarDeposito.** Recibe como argumento un tipo de pintura, una cantidad en litros (menor de 1000 litros) y un precio por litro y lo asigna a un depósito. El depósito usado se devuelve como resultado del subprograma y será preferiblemente un depósito que ya contiene ese tipo de pintura (y el precio se calcula como la media del precio anterior y el nuevo) en el que cabe la cantidad recibida. Si la cantidad cargada no cabe en ningún depósito con ese mismo producto entonces se cargará en un depósito vacío. Si no existe opción de almacenar la pintura se devuelve un cero.
- **ListarStockCompleto.** Muestra un listado de diez líneas en el que se identifica cada depósito, el tipo de pintura almacenada, el precio por litro actual y la cantidad almacenada en ese momento.
- **RealizarOferta.** Recibe como parámetros un tipo de pintura y una cantidad en litros pedidos y devuelve el depósito con la mejor oferta disponible con los precios almacenados en cada depósito. Si no se puede realizar una oferta por falta de disponibilidad se devolverá como depósito de resultado un cero. Para simplificar el problema, considere que no pueden combinarse pintura del mismo tipo de distinto depósito para una oferta. Una oferta hace referencia exclusivamente a un depósito.