

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 preguntas 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. En C±, los vectores abiertos:
 - A. Sólo pueden utilizarse en la definición de funciones
 - B. Sólo pueden utilizarse en la definición de procedimientos
 - C. Pueden utilizares en la definición de funciones y en la definición de procedimientos
 - D. No pueden utilizarse ni en la definición de procedimientos ni en la definición de funciones
2. El siguiente fragmento de código en C±:


```
x = 2;
do{
    printf("%d ", x);
    x = x+1*2/3-1;
}while(x > 0);
```

 - A. Imprime: 2
 - B. Imprime: 2 1
 - C. Imprime: %d
 - D. Entra en un bucle infinito
3. La técnica de desarrollo mediante refinamientos sucesivos consiste en:
 - A. Descomponer acciones en otras más simples en pasos sucesivos
 - B. Depurar los errores del código en pasos sucesivos
 - C. Reducir el tiempo de compilación de un programa en pasos sucesivos
 - D. Reducir el tiempo de ejecución de un programa en pasos sucesivos
4. El esquema de datos formación se corresponde con el esquema de acción:
 - A. Secuencia
 - B. Iteración
 - C. Selección
 - D. Tupla
5. Dadas las variables A y B de tipo vector de tres elementos, si queremos copiar todos los elementos de A en B, en C± escribimos:
 - A. B=A;
 - B. for(int i=1; i<=3; i++){ B[i]=A[i]; }
 - C. A=B;
 - D. for(int i=0; i<3; i++){ B[i]=A[i]; }
6. En C±, dentro de un subprograma, un argumento:
 - A. No puede cambiar de tipo
 - B. Puede cambiar de tipo si se pasa por referencia
 - C. Puede cambiar de tipo si se pasa por valor
 - D. Nunca puede cambiar de valor
7. ¿Qué características tienen los elementos locales?
 - A. Son visibles en su ámbito y fuera
 - B. Son invisibles en su ámbito y fuera
 - C. Son invisibles en su ámbito y visibles fuera
 - D. Son visibles en su ámbito e invisibles fuera
8. El modelo de programación lógica:
 - A. Imita la arquitectura de Von Neumann
 - B. Se basa en el uso de funciones
 - C. Se basa en la declaración de hechos y reglas
 - D. Es el que utiliza C±
9. En C±, cuando se utiliza:


```
typedef struct Uno{Dos Tres; Cuatro Cinco};
```

 - A. Uno y Cuatro son identificadores de tipo
 - B. Tres y Cinco son identificadores de tipo
 - C. Dos y Cinco son identificadores de tipo
 - D. Uno y Tres son identificadores de tipo
10. Tras la ejecución del siguiente fragmento de código en C±:


```
int a;
void proc(int & b, int & c){
    b = 2;
    c = a%2;
    b = a+b*c;
}
...
a = 1;
proc(a,a);
printf("Variable a = %d", a);
```

 - A. La variable a vale 3
 - B. La variable a vale 2
 - C. La variable a vale 1
 - D. La variable a vale 0

EJERCICIO DE PROGRAMACIÓN

Realizar en C± un TAD, con fichero de interfaz y de implementación, capaz de almacenar hasta 50 valores enteros. El TAD dispondrá de un único subprograma: **Moda**, que devuelve la moda de los números enteros almacenados en el TAD. Por ejemplo, la moda de {1,6,2,5,4,2,1,3,4,2} es 2 porque es el valor que aparece más veces (3 veces). En caso de que haya varias modas, se devolverá una cualquiera de ellas.