

FUNDAMENTOS DE PROGRAMACION	MODELO 1	CONV. ORD. 21/22
GRADO DE INFORMÁTICA Y GRADO DE TECNOLOGÍAS DE LA INFORMACIÓN-ETSII-UNED		
MATERIAL AUTORIZADO: NINGUNO		DURACIÓN: 2 HORAS
INSTRUCCIONES:	<p>1) El test debe ser contestado en la hoja de marcas. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta.</p> <p>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.</p> <p>3) Cada respuesta correcta 1 pto. Respuesta incorrecta, doble o en blanco: 0 pto.</p> <p>4) Puede quedarse, si lo desea, con esta hoja de examen.</p>	

CUESTIONES DE TEST

1. En el lenguaje C++, un procedimiento acaba:

- A. Cuando se han ejecutado todas sus sentencias
- B. Cuando ejecuta su última sentencia o una sentencia return
- C. Sólo cuando ejecuta una sentencia return
- D. Sólo cuando ejecuta la última sentencia

2. El siguiente código:

```
char i = '9';
int op = 0;

for (int j=10; j>=1; j--) {
    op = j+(int(i)-int('1'))/2;

    if (op%3==1) {
        printf("%d ", j);
        i = char(int(i)-1);
    }
}
```

Imprime:

- A. 8 5 3 1
- B. 9 7 4 2
- C. 9 6 3
- D. 8 6 4 2

3.- La corrección de un programa:

- A. Es parcial si produce un resultado correcto más de la mitad de las veces que termina
- B. Es parcial si produce un resultado correcto en caso de que termine
- C. Es parcial si siempre termina, aunque a veces genere resultados incorrectos
- D. Nunca puede ser total

4.- El diagrama de estructura modular refleja:

- A. El acoplamiento y la cohesión
- B. El encapsulado de elementos
- C. La reutilización de tipos y subprogramas
- D. Los módulos y sus relaciones

5.- El modelo abstracto de cómputo de flujo de datos se basa en el empleo de:

- A. Redes de operadores
- B. Funciones
- C. Hechos y reglas
- D. Secuencias de órdenes

6.- El siguiente código:

```
void Procedimento(int x, int &y) {
    if (x>=y) {
        y = 2*x;
    } else {
        x = y-1;
    }
}

int main()
{
    int x = 2;
    int y = 2;
    Procedimento(x, y);
    printf("%d %d", x, y);
}
```

Imprime:

- A. 2 2
- B. 2 4
- C. 4 2
- D. 4 4

7.- En el lenguaje C++ para realizar una estructura de datos no acotada es necesario utilizar:

- A. Formaciones
- B. Tipos Abstractos de Datos (TADs)
- C. Punteros
- D. Tablas

8.- En el lenguaje C++ y respecto a cuándo se puede utilizar el elemento:

private:

Se puede afirmar lo siguiente:

- A. Se puede declarar múltiples veces dentro de un tipo registro
- B. Se puede utilizar múltiples veces dentro de cualquier declaración de datos
- C. Sólo se puede declarar una vez dentro de un tipo registro
- D. Se puede utilizar múltiples veces en cualquier punto de un programa

9.- En C++:

- A. Se pueden mezclar declaraciones y sentencias ejecutables
- B. La parte ejecutiva precederá a la parte declarativa
- C. Nunca se puede mezclar declaraciones y sentencias ejecutables
- D. Solo se agruparán sentencias en el uso de subprogramas

10.- Dado el siguiente código en C++:

```
int main(){
    int cont = 0;
    for (int i=1;i<4;i++) {
        for (int j=3; j>=0; j--) {
            cont = (i+j) % 4;
            if ((cont%(j+1))==1)
            {
                continue;
            }
        }
        if ((cont%4)==1)
        {
            continue;
        }
    }
}
```

¿Cuál sería el valor final de la variable cont?

- A. 3
- B. 2
- C. 0
- D. 1

EJERCICIO DE PROGRAMACIÓN

Un ambulatorio presta servicio hasta 1000 pacientes.

Realice un Tipo Abstracto de Datos (TAD) que almacene la siguiente información de los pacientes:

- Nombre
- Primer apellido
- Segundo apellido
- ¿Vacunado para el COVID? Sí o No
- Tipo de la vacuna: Pfizer, Moderna, AstraZeneca o Janssen
- Número de dosis: 1, 2, 3

El TAD dispondrá de los siguientes subprogramas:

- **PorcentajeNoVacunados**, que indique qué porcentaje de pacientes no se ha vacunado.
- **ListaJanssen**, que muestra los datos (nombre y apellidos) de los pacientes vacunados con Janssen.
- **NumeroMedioDosis**, que muestre el número medio de dosis por paciente según el tipo de la vacuna.