

## GUÍA PARA EL EXAMEN DE FUNDAMENTOS DE PROGRAMACIÓN

**Problema 1.** Elija la respuesta correcta

1. La administración de recursos de almacenamiento es parte de:

- a) Computadora
- b) Compilador
- c) Sistema Operativo
- d) Internet

2. ¿Qué es un algoritmo?

- a) Es un proceso que se ejecuta en una computadora
- b) Es una secuencia de pasos que resuelve un problema en un tiempo finito
- c) Es un programa en Java
- d) En un proceso que ejecuta una serie de pasos con un fin.

3. ¿Cuál es la diferencia entre un algoritmo y un proceso?

- a) Ninguna son lo mismo
- b) Los procesos se codifican en lenguajes como C, Java o C++ y los algoritmos son conceptuales.
- c) Los algoritmos tienen la garantía de poderse codificar y los procesos no.
- d) Los algoritmos tienen la garantía que son finitos y los procesos no.

4. ¿Cuál es la diferencia entre un algoritmo y un programa?

- a) Ninguna son lo mismo
- b) Los programas se codifican en lenguajes como C, Java o C++ y los algoritmos son conceptuales.
- c) Los algoritmos tienen la garantía de poderse codificar y los programas no.
- d) Los algoritmos tienen la garantía que son finitos y los programas no.

5. Son tareas de un compilador

- a) Analizar léxicamente un archivo escrito en algún lenguaje de programación.
- b) Analizar sintácticamente un archivo escrito en algún lenguaje de programación.
- c) Generar código ejecutable.
- d) Todas las anteriores.

6. ¿Cuál es las diferencias entre un compilador y un intérprete?

- a) El compilador genera código ejecutable y el intérprete analiza cada línea del programa y lo va ejecutando.
- b) Ninguna, son lo mismo.

- c) El compilador genera bibliotecas de funciones y el intérprete las ejecuta.
- d) El compilador corre los programas fuentes en tiempo real y el intérprete usa una máquina virtual.

7. ¿Qué es un paradigma de programación?

- a) Es un conjunto de conceptos que dan un marco de referencia para diseñar programas.
- b) Es un conjunto de instrucciones para codificar un algoritmo.
- c) Es un modelo para correr programas.
- d) Es una forma de documentar programas.

8. No es un paradigma de programación.

- a) Programación funcional.
- b) Programación orientada a objetos.
- c) Programación estructurada.
- d) Programación vectorizada.

**Problema 2.** Los lenguajes de programación de alto nivel contienen sentencias IF para condicionales y WHILE para ciclos. En ocasiones los lenguajes de programación contienen sentencias BREAK para romper. Considere el siguiente pseudo código

```
n := 100
WHILE n > 1
{
    a := funcion( n );
    IF a < 0 THEN
    {
        BREAK;
    }
    n := n-1
}
```

Escriba el código anterior sin usar la sentencia IF.

**Problema 3.** Los lenguajes de programación de alto nivel contienen sentencias WHILE y FOR para control de ciclos. Considere el siguiente pseudo código

```
n := 1000
x := 0
dx := 3.1416/n
FOR i = 1 TO n
{
  a[i] := sin(X)
  x := x + dx
}
```

Escriba el código anterior usando la sentencia WHILE.

**Problema 4.** Los lenguajes de programación de alto nivel contienen sentencias IF e IF-ELSE para condicionantes. Considere el siguiente pseudo código

```
a := RANDOM() // La función RANDOM genera un número entre -1 y 1
IF a > 0 THEN {
  PRINT "NUMERO POSITIVO"
}
ELSE IF a < 0 THEN {
  PRINT "NUMERO NEGATIVO"
}
ELSE {
  PRINT "CERO"
}
```

Escriba el código anterior únicamente usando sentencias WHILE.

**Problema 5.** El sistema sexagesimal es un sistema que divide cada unidad en 60 unidades de orden inferior, es decir, es un sistema en base 60. Este sistema lo podemos ver reflejado en la medida del tiempo, pero también es ampliamente utilizado para medir la amplitud de los ángulos donde las unidades de medición son los grados, minutos y segundos como coordenadas terrestres.

Se puede pasar un número decimal a sexagesimal de la siguiente forma. Como ejemplo convertiremos el valor decimal  $121.135^\circ$  al sistema sexagesimal de la siguiente manera:

1. La parte entera de los grados decimales será también los grados del sistema sexagesimal ( $121^\circ$ ).
2. Multiplicar los decimales del valor original por 60 ( $0.135 * 60 = 8.1$ ).
3. La parte entera del resultado anterior serán los minutos ( $8'$ ).
4. Multiplicar los decimales del paso 2 por 60 ( $0.1 * 60 = 6$ ).
5. El resultado del paso anterior serán los segundos ( $6''$ ).
6. Tomar los resultados de los pasos 1, 3 y 5 para formar el resultado final ( $121^\circ 8' 6''$ ).

Escriba un programa en pseudo código que convierta un número decimal a sexagesimal.

**Problema 6.** Considere los números en sistema sexagesimal del Problema 5. Escriba un programa en pseudo código que pase números en sistema sexagesimal a números decimales.

**Problema 7.** Considere un arreglo de  $N$  elementos donde no hay elementos iguales.

Escriba en pseudo código el algoritmo de ordenación Burbuja.

**Problema 8.** Considere un arreglo de  $N$  elementos donde no hay elementos iguales.

Escriba en pseudo código el algoritmo de ordenación Quick Sort.

**Problema 9.** Considere un arreglo de  $N$  elementos donde no hay elementos iguales.

Escriba en pseudo código el algoritmo de ordenación Shell Sort.

**Problema 10.** Una serie de Fibonacci hace referencia a la secuencia

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ....

entonces el  $n$ -ésimo elemento de la serie se puede calcular como

$$f_n = f_{n-1} + f_{n-2}$$

Escriba en pseudo código una función recursiva que calcule los primeros  $N$  números de la serie de Fibonacci.

**Problema 11.** Escriba en pseudo código una función iterativa que calcule los primeros  $N$  números de la serie de Fibonacci.

**Problema 12.** Una lista es una estructura FIFO, donde el primero en entrar es el primero en salir. Cuando la lista se implementa con un arreglo de elementos se dice que se tiene una cola de tamaño finito. Una pila creada sobre un arreglo tiene índices que se mueven indicando el inicio y fin de la fila. La pila tiene un conjunto de funciones:

- a) INIT ( array ) para inicializar la pila
- b) PUT ( element, array ) agregar elemento.
- c) GET ( array ) para obtener el siguiente elemento disponible de la cola.
- d) STATUS ( array ) para conocer el estado de la pila: EMPTY cuando la cola vacía), FULL cuando la cola llena, y AVAILABLE cuando la cola tenga elementos pero aún se pueden almacenar nuevos elementos.

Escriba en pseudo código las funciones INIT, PUT, GET y STATUS de una cola implementada en un arreglo.

**Problema 13.** La pila es una estructura LIFO, donde el último elemento almacenado es el primero en salir. Una pila se puede implementar usando una lista ligada. Las pilas tienen las siguientes funciones:

- a) PUSH ( elemento, lista ) para meter un elemento a la pila.
- b) POP ( lista ) para sacar un elemento de la pila.
- c) STATUS ( lista ) para obtener el estado de la pila: EMPTY o NON EMPTY.

Escriba en pseudo código las funciones PUSH, POP y STATUS para una pila implementada en una lista. NOTA: no use listas doblemente ligadas.

