**VERDADERO y FALSO**

1. Un algoritmo es un conjunto infinito de operaciones que permiten resolver un tipo de problema. FALSO.

-Debería ser finito

2. El concepto de algoritmo solo se aplica a problemas que serán resueltos por computadoras. FALSO

-Se puede aplicar en la vida cotidiana

3. Un algoritmo debe ser aplicable a un problema específico y no a un tipo de problemas. FALSO.

-Ya que una de las características de los algoritmos es que debe ser general y no especifico.

4. Las recetas de cocina son ejemplos de algoritmos. VERDADERO.

5. Un algoritmo siempre debe ser enunciado en un lenguaje de programación. FALSO

-Un algoritmo puede ser enunciado hasta en el lenguaje humano, como una receta de cocina.

6. La eficacia de un algoritmo no depende de los recursos con los que se cuente. FALSO

-Si depende porque los recursos son importantes para desarrollar un mejor programa, con mayor eficacia y versatilidad.

7. Para cualquier problema que pueda especificarse un método finito de solución, puede definirse un algoritmo. VERDADERO

8. Un ejecutante es una entidad capaz de entender y ejecutar los pasos descritos en un algoritmo. VERDADERO

9. Una acción no-primitiva no puede ser descompuesta en acciones primitivas. FALSO

-Si se puede, ya que una acción no primitiva ya viene incluida con acciones primitivas.

10. La programación modular es un método de diseño que divide un problema en partes independientes. VERDADERO

11. La programación modular disminuye la complejidad del problema original. VERDADERO

12. Un módulo en la programación modular debe tener múltiples entradas y salidas. FALSO

-Ya que un módulo puede tener simplemente una entrada y salida, no tener, o tener múltiples.

13. La programación modular facilita las modificaciones y conexiones en la resolución de un problema. VERDADERO

1: ¿Cuáles son las etapas principales para la resolución de problemas computacionales?

R: Las etapas principales para la resolución de problemas computacionales son:

Definición, Análisis, Programación y Prueba.

2: ¿Qué tres conjuntos se deben identificar durante el análisis de un problema computacional?

R: Durante el análisis de un problema computacional, se deben identificar los conjuntos de Datos, Resultados y Condiciones.

3: ¿Cuáles son las dos subetapas que componen la etapa de programación en la resolución de problemas computacionales?

R: Son la de elección y creación del método y codificación.

4: ¿Qué es una estrategia en el contexto de la resolución de problemas computacionales y cómo se diseña?

R: Una estrategia en el contexto de la resolución de problemas computacionales es un plan que indica las tareas a realizar para resolver el problema. Se diseña mediante el análisis detallado del problema y la identificación de las operaciones necesarias para transformar las entradas en las salidas deseadas.

5: ¿Qué es el diseño descendente (top-down) y cuáles son sus características generales?

R: El diseño descendente (top-down) es un enfoque en el que se inicia con el problema general y se va descomponiendo en subproblemas más pequeños y manejables. Sus características generales incluyen la división de tareas complejas en tareas más simples y la resolución de problemas de manera sistemática.

6: ¿Por qué es importante la definición correcta y completa del problema en la resolución de problemas computacionales?

R: Es importante porque proporciona una comprensión clara del problema, lo que permite desarrollar una solución adecuada y eficaz.

7: ¿Cuál es el objetivo de la etapa de prueba en la resolución de problemas computacionales?

R: El objetivo de la etapa de prueba en la resolución de problemas computacionales es verificar que la solución desarrollada funcione como se espera y produzca los resultados deseados.

8: ¿Qué se entiende por "mejor alternativa" al elegir un método para resolver un problema computacional?

R: Al elegir un método para resolver un problema computacional, "mejor alternativa" se refiere a la opción que resuelve el problema de manera más eficaz y eficiente, considerando aspectos como la simplicidad, la eficiencia en términos de tiempo y espacio, y la adaptabilidad a cambios futuros.

9: ¿En qué consiste la codificación durante la etapa de programación en la resolución de problemas computacionales?

R: La codificación durante la etapa de programación en la resolución de problemas computacionales consiste en la implementación del algoritmo diseñado en un lenguaje de programación específico.

10: ¿Cuál es el propósito de dividir un problema en subproblemas más simples en el diseño de una estrategia?

R: El propósito de dividir un problema en subproblemas más simples en el diseño de una estrategia es hacer el problema más manejable y fácil de resolver, permitiendo abordar cada parte del problema individualmente antes de combinar las soluciones para obtener la solución completa.

11: ¿Qué es un algoritmo y cuáles son sus características principales?

R: Un algoritmo es un conjunto finito de operaciones que, seguidas en un determinado orden, permiten resolver un tipo de problema. Sus características principales son ser finito, definido y general.

12: ¿Qué es un "ejecutante" en el contexto de la resolución de problemas computacionales?

R: Un "ejecutante" es una entidad capaz de entender un enunciado y ejecutar los pasos descriptos en un algoritmo.

13: ¿Cuál es la diferencia entre una acción primitiva y una acción no-primitiva en un algoritmo?

R: Una acción es primitiva cuando su enunciado es suficiente para ser ejecutada sin información adicional. Una acción no-primitiva puede ser descompuesta en acciones primitivas para un ejecutante dado.

14: ¿Qué es una condición en el contexto de un algoritmo?

R: Una condición es una afirmación lógica sobre el estado de algún recurso del ambiente que puede tomar valor verdadero o falso en el momento de la observación.

15: ¿Qué es la programación modular y cuáles son sus objetivos?

R: La programación modular es un método de diseño que divide el problema en partes independientes. Sus objetivos son disminuir la complejidad, aumentar la claridad, aumentar la confiabilidad y facilitar modificaciones y conexiones.

16: Mencione un ejemplo de un algoritmo en la vida diaria.

R: Un ejemplo de un algoritmo en la vida diaria puede ser una receta de cocina.

17: ¿La eficacia de un algoritmo depende de los recursos y factores considerados? ¿Por qué?

R: Si, la eficacia de un algoritmo depende de los recursos y factores considerados porque estos pueden influir en el costo y el tiempo de ejecución del algoritmo.

18: ¿Qué es el ambiente del problema en el contexto de la resolución de problemas computacionales?

R: El ambiente del problema es el conjunto de recursos existentes en el momento de la ejecución de un trabajo.

19: ¿Cuál es el propósito de dividir un problema en módulos en la programación modular?

R: Dividir un problema en módulos facilita su análisis, resolución y puesta a punto, al poder tratar cada parte de manera individual.

20: ¿Qué criterios se deben tener en cuenta para determinar un módulo en la programación modular?

R: Para determinar un módulo en la programación modular se deben tener en cuenta criterios como que el módulo corresponda a una función lógica bien definida, que sea de poca complejidad, que tenga una estructura de caja negra y que tenga una única entrada y salida.

21: ¿Qué es una constante en un algoritmo computacional?

R: Una constante en un algoritmo computacional es un elemento cuyo valor no puede alterarse durante la ejecución de dicho algoritmo.

22: ¿Qué es una variable en un algoritmo computacional?

Es un espacio de memoria reservado para almacenar un valor. La variable tiene un nombre que se utiliza para hacer referencia a ese espacio de memoria y puede contener diferentes valores durante la ejecución del programa.

23. ¿Cuáles son las tres reglas simples para proponer el nombre de una variable en un algoritmo?

R: Las tres reglas son:

a) utilizar sólo letras y/o dígitos, comenzando siempre con una letra.

b) no utilizar palabras claves o reservadas en el pseudocódigo.

c) no hacer distinción entre mayúsculas y minúsculas.

24. ¿Qué es una expresión en un algoritmo computacional?

R: Una expresión es un conjunto de operandos ligados por operadores que describen una operación o cálculo arrojando un único resultado.

25. ¿Cuáles son los tres tipos de datos que un algoritmo computacional puede emplear?

R: Los tipos de datos son: Numérico, carácter y Lógico

26. ¿Cómo se clasifican los datos de tipo numérico?

R: Los datos de tipo numérico se clasifican en tipo entero o tipo real.

27. ¿Qué es una asignación en un algoritmo computacional y cuáles son los tipos?

R: Una asignación en un algoritmo computacional es una acción que permite a un identificador o variable, representar o memorizar cierto valor.

Una asignación puede ser:

A. Aritmética

B. Carácter

C. Lógica

28. ¿Qué son las primitivas en un algoritmo computacional y como se clasifican?

R: Son acciones cuyo enunciado es suficiente para que el ejecutante pueda realizarla sin ningún tipo de información adicional.

Se clasifican en

A. Secuenciales

B. Condicionales

C. Repetitivas

Verdadero y Falso

1. Una constante puede cambiar su valor durante la ejecución de un algoritmo. FALSO

-Ya que al asignar el valor a una constante, durante la ejecución del algoritmo no puede cambiar.

2. Las variables en un algoritmo pueden cambiar su valor durante la ejecución. VERDADERO

3. Las variables pueden ser nombradas con palabras claves o reservadas en el pseudocódigo. FALSO

-No se puede, ya que hay distintas palabras las cuales ya están reservadas en el lenguaje.

4. Una expresión describe una operación o cálculo arrojando múltiples resultados. FALSO

-Arroja un único resultado

5. Los tipos de datos que un algoritmo computacional puede emplear son solo

Numérico y Caracter. FALSO

-También pueden ser lógicos, o reales.

6. Los datos de tipo numérico se clasifican en tipo entero y tipo complejo. FALSO

-Ya que se clasifican en tipo entero y real.

7. Una variable numérica puede representar un dato no numérico. FALSO

- No porque si le pones “xd” no lo almacena.

8. Una asignación en un algoritmo computacional permite a un identificador o variable, representar o memorizar cierto valor. VERDADERO

9. Las primitivas en un algoritmo computacional son acciones que requieren información adicional para ser ejecutadas. FALSO

-Las primitivas en un algoritmo computacional son acciones básicas y elementales que no requieren información adicional para ser ejecutadas

1. ¿Qué es la diagramación estructurada?

R: La diagramación estructurada es una metodología que establece que todo algoritmo, sin importar su complejidad, puede elaborarse mediante el uso de solo tres estructuras lógicas de control: secuencia, selección e iteración.

2. ¿Cuáles son las tres estructuras lógicas de control según la diagramación estructurada?

R: Las tres estructuras lógicas de control son: secuencia, selección e iteración.

3. ¿Cómo se define una estructura de secuencia?

R: La estructura de secuencia es un conjunto de acciones que se ejecutan en el mismo orden en que aparecen en el algoritmo, una a continuación de la otra.

4. ¿Qué permite la estructura de selección en un algoritmo?

R: La estructura de selección permite alterar el flujo lineal de control de las acciones de un algoritmo. Proporciona la posibilidad de tomar decisiones y elegir un camino a seguir basándose en ciertos valores específicos del algoritmo

5. ¿Cuántos tipos de estructuras de selección existen y cuáles son?

R: Existen dos tipos de estructuras de selección: la estructura condicional o de decisión y la estructura de selección múltiple.

6. ¿Qué es una estructura condicional anidada?

R: En una estructura condicional anidada, tanto la alternativa verdadera como la falsa pueden presentar a su vez, otras estructuras condicionales.

7. ¿Qué son las estructuras de control complejas en el diseño de algoritmos?

R: Las estructuras de control complejas en el diseño de algoritmos son aquellas que se basan en la combinación de estructuras elementales. Un ejemplo son los condicionales anidados, que son una estructura de decisión dentro de otra estructura de decisión.

1. Todo algoritmo, por más complejo que sea, puede elaborarse mediante el uso de solo tres estructuras lógicas de control. VERDADERO

2. La estructura de secuencia implica que las acciones se ejecutan en un orden aleatorio. FALSO

-En la estructura de secuencia se ejecutan paso a paso según el algoritmo que se diseñó.

3. La estructura de selección permite alterar el flujo lineal de control de las acciones de un algoritmo. VERDADERA (si, según)

4. Las estructuras de selección solo permiten elegir entre dos alternativas. FALSO

- La estructura de selección según, deja más de dos opciones.

5. En una estructura condicional anidada, solo la alternativa verdadera puede presentar otras estructuras condicionales. FALSO

- Ya que tanto como en la alternativa verdadera o falsa se pueden presentar otras estructuras condicionales.