**Programación I**

1. ¿Qué es un algoritmo computacional?
2. ¿Cuáles son las características que debe reunir un algoritmo computacional?
3. ¿Cuáles son los elementos de solución de un problema?
4. ¿Cuál es el criterio para determinar si un dato es de entrada?
5. ¿Qué es el proceso?
6. ¿Qué es una salida?
7. ¿Cuáles son las etapas básicas de la metodología de programación?
8. ¿Para que utilizamos los diagramas de flujo?
9. Describe los símbolos utilizados en el diagrama de flujo
10. ¿Qué es un pseudocódigo?
11. ¿Qué es código?
12. ¿Qué es un lenguaje de programación?
13. ¿Cuál es la diferencia entre programa fuente y programa ejecutable?
14. ¿Qué es una variable?
15. ¿Cómo se declaran las variables?
16. ¿Cuáles son las reglas para poner nombre o definir una variable?
17. ¿Qué es una constante?
18. ¿Cómo se declaran las contantes?
19. ¿Qué son los tipos de datos?
20. ¿Cuáles son los tipos de datos numéricos en C?
21. ¿Cómo se elige el tipo de dato que debe ser una variable?
22. ¿Qué utilidad tienen los operadores aritméticos?
23. Menciona los operadores aritméticos
24. Menciona cuáles son los operadores de asignación
25. ¿Qué es una expresión aritmética?
26. Explica la jerarquía de los operadores aritméticos
27. ¿Cuál es la función de los paréntesis en una expresión aritmética?
28. ¿Qué son las librerías o bibliotecas de un lenguaje de programación?
29. ¿Cuál es la utilidad de la sentencia printf?
30. ¿Cuál es la utilidad de la sentencia scanf?
31. ¿Qué es un compilador y cuáles son sus funciones?
32. ¿Qué es la sintaxis de un leguaje?
33. ¿Cómo identificamos un error de sintaxis?
34. ¿Qué es un error de semántica?
35. ¿Qué es un error de lógica?
36. ¿Qué o quién es un usuario?

**Respuestas**

1. es una secuencia de pasos que se deben llevar a cabo para resolver una tarea específica.

2. debe ser preciso:

Definido:

Finito:

3. entrada, proceso y salida.

4. datos que se necesitan para realizar al proceso, se pueden solicitar al usuario o no.

5. son los pasos que se deben llevar a cabo para resolver un problema.

6. son los resultados del proceso.

7. definición del problema, análisis del problema, diseño de la solución, codificación, prueba y depuración y documentación.

8. nos sirven como una representación gráfica del algoritmo.

9. óvalo para el inicio y final, rombo para decisiones, rectángulo para el proceso, círculo para el inicio del programa, romboide para la entrada, hoja impresa para salida y líneas para especificar el orden.

10. literalmente significa “falso lenguaje”, ya que es un lenguaje para especificar algoritmos que sigue pasos determinados para resolver un problema.

11. es un conjunto de instrucciones contenidas en un programa.

12. conjunto de sentencias usadas por los programadores para escribir secuencias de instrucciones que ejecuta una computadora.

13. el código fuente es el conjunto de instrucciones que son escritas en un lenguaje de programación, y el código ejecutable es el conjunto de códigos que ha sido compilado y que ejecuta la computadora.

14. son datos que el programa guarda pero que pueden ser modificados a lo largo de la ejecución de un problema.

15. tipo de variable y un identificador.

16. no deben ser nombradas como una “palabra reservada” del lenguaje, solo pueden ser letras, dígitos, el guión y guión bajo, debe empezar con un carácter y nunca nombrar dos variables globales con el mismo nombre.

17. es un valor que no puede ser alterado durante la ejecución de un programa.

18. en C++ se declaran con la directiva #define

19. enteros: byte, short, int y long. Flotantes: float y double. Booleanos y Char

20. entero, entero largo, entero corto, decimal simple y decimal doble.

21.depende del tipo de dato que dependa el problema

22. para realizar cálculos aritméticos si se requiere para el programa.

23. suma, resta, multiplicación, división y potencialización%.

24. son una forma de realizar una operación y una asignación al mismo tiempo

25., el resultado es un número y los operadores son aritméticos, módulos, residuo, paréntesis potencias, multiplicación y división y de ultimo suma y resta.

26: primero van las potencias, luego los paréntesis, luego la multiplicación, división y módulo, y por último la suma y la resta.

27. estos sirven para que se realice primero la operación que esta entre ellos.

28. son el conjunto de recursos que pueden ser utilizados por el programador para realizar diversas operaciones.

29. para la salida de datos

30. para guardar los datos de entrada en las variables.

31. es un programa informático que traduce un programa de lenguaje de programación a lenguaje común, su función es para traducir el lenguaje a uno más sencillo para que lo entienda la máquina e informa al usuario si existe algún error en el código fuente.

32. es el conjunto de reglas que se siguen para que el compilador reconozca el programa como válido.

33. escribir mal una funcion

34. es cuando el significado no es el que se quería, porque se entendió mal el problema o no se entendió completamente.

35. es el tipo de error que se produce cuando los resultados no son los esperados, pero no por el compilador, sino por el programador que no realizó bien alguna operación o la realizó sin el conocimiento necesario.

36. es un conjunto de recursos o permisos a los que se tiene acceso, concretamente en programación es la persona a la que va destinado un producto cuando éste ya superó las etapas de desarrollo.