

AUTOEVALUACIÓN PAGINA 110.

- 1 ¿Qué son las estructuras selectivas?
R//Son aquellas que dirigen la ejecución de un programa hacia un grupo de sentencias del resultado de la condición.
- 2 ¿Cuándo se ejecuta la sentencia de if?
R// El if se usa para tomar decisiones, el if evaluará una operación lógica que de como resultado true o false, y lo más importante que se ejecuta la pieza de código siguiente siempre y cuando el resultado sea verdadero.
- 3 ¿Qué colocamos entre paréntesis en if?
R// Una expresión a evaluar.
- 4 ¿Qué es una expresión relacional?
R// Son símbolos que se usan para comparar dos valores. Si el resultado de la comparación es correcto la expresión considerada es verdadera, en caso contrario es falsa.
- 5 ¿Cuáles son los operadores de una expresión relacional?
R//
< menor que
> mayor que
== igual a
~= no igual a
<= menor que o igual a
>= mayor que o igual a
- 6 ¿A qué valores posibles puede evaluar una expresión relacional o lógica?
R// A los valores true o false.
- 7 ¿Qué es una expresión lógica?
R// Es una expresión que sólo puede tomar estos dos valores, verdadero y falso.
- 8 ¿Cuáles son los operadores de las expresiones lógicas?
R//
&& y
|| o
! no
- 9 ¿Qué operador solamente necesita un operando?
R// El operador de la negación.

10 ¿Cómo funciona el switch?

R// transfiere el control a un conjunto de una o más sentencias de asignación de reglas, dependiendo del valor de una expresión. Este caso se ejecuta si la expresión coincide con alguna de las series especificadas. La sentencia switch sólo comprueba coincidencias exactas.

11 ¿Qué es la variable de comparación y cómo se coloca?

R// Es la variable excluyente, las variables a comparar se ubican en las columnas y en las filas, se ponen los valores que toman las variables por rangos o categorías.

12 ¿Cómo definimos los casos y cómo usamos break?

R// Para indicar un caso, usamos case seguido del valor de comparación y dos puntos.

R// El break lo usamos para terminar el caso si no lo hacemos así el compilador nos puede dar problemas.

AUTOEVALUACIÓN PAGINA 146.

- 1 ¿Qué es un ciclo?
R// Es una secuencia de instrucciones de código que se ejecuta repetidas veces, hasta que la condición asignada a dicho bucle deja de cumplirse.
- 2 ¿Cuáles son las partes del ciclo for?
R// - Inicialización, asignación, condición, código y por último incremento.
- 3 ¿Cómo colocamos el valor inicial de conteo en un ciclo for?
R// La sentencia del ciclo se inicia con la palabra clave for seguida de paréntesis. Adentro de los paréntesis colocaremos expresiones que sirven para controlar el ciclo.
- 4 ¿Cómo colocamos el valor final de conteo en un ciclo for?
R// En esta última línea hacemos uso de la llave cerrando "}", una vez más como seguramente ya sabrás esta nos indica que allí termina el bloque del ciclo for y se dará por terminada la ejecución de este para continuar ejecutando el resto del algoritmo.
- 5 ¿Cómo se lleva a cabo el incremento en un ciclo for?
R// Nuestro incremento ha sido de uno en uno. Sin embargo, podemos hacer que el conteo sea de dos en dos, de tres en tres o de cualquier otro valor. Para lograr esto, simplemente tenemos que modificar el incremento e indicar cómo se incrementaría nuestra variable de control. El valor del incremento puede ser un valor colocado explícitamente o el valor que se encuentra adentro de una variable.
- 6 ¿Cómo funciona el ciclo do while?
R// Permite que cierto código se repita mientras una condición se evalúe como verdadera. El valor de la evaluación dependerá del estatus del programa en un momento dado.
- 7 ¿Por qué el ciclo while se lleva a cabo al menos una vez?
R//
- 8 ¿Se necesita punto y coma al finalizar el ciclo do while?
R// Si
- 9 ¿Cómo funciona el ciclo while?
R// Nos permite repetir el código un número desconocido de veces, pero en este caso el ciclo puede o no puede ejecutarse dependiendo de su condición
- 10 ¿Cuántas veces se puede repetir el ciclo while?
R// Un número indefinido de veces.
- 11 ¿Qué tipo de condición podemos colocar en el ciclo while?
R// El propósito es repetir un bloque de código mientras una condición se mantenga verdadera. sí, verdadero (true) si la condición se cumple, o falso si esta no se cumple (false). También puede contener el nombre de una variable booleana, y el valor de la expresión

dependerá de su contenido. Se debe tener en cuenta que además de las variables también puede haber llamadas a funciones que devuelvan un valor.

12 ¿Se coloca un bloque de código en el ciclo while?

R// si