

Clase VI:

Loops infinitos, sentencias break y continue

Repaso general

Informática I
Centro Regional Universitario Córdoba
UNDEF

11 de mayo de 2021

Condiciones infinitas, sentencias break y continue

Condiciones infinitas

Cuando en un ciclo repetitivo controlado por centinela la condición a evaluar es siempre verdadera, por ejemplo $1=1$ o $2!=0$, estamos en presencia de un ciclo de repetición infinito.

Sentencia break

Cuando una sentencia break se encuentra dentro de un bucle, este termina su ejecución y el programa continua con las líneas de código que sucedan al ciclo repetitivo.

Sentencia continue

Cuando una sentencia continue se encuentra dentro de un ciclo repetitivo, el programa regresa al comienzo del mismo, evitando todas las líneas que se encuentren después de haber encontrado la sentencia mencionada.

Sentencias break y continue: ejemplos

- 1 Diseñar y codificar un programa que calcule la suma de 10 números ingresados por teclado. Si el operador ingresa algún número par, el programa finalizar la toma de datos e imprimir la suma.

► [Ver en github](#)

- 2 Diseñar y codificar un programa que calcule la suma de 10 números ingresados por teclado. Los números pares no deben ser contemplados en la suma.

► [Ver en github](#)

- 3 Diseñar y codificar un programa que calcule la suma de todos los números ingresados por teclado hasta que el usuario ingresa la opción -10. Además, el numero 100 no debe ser contemplado en la suma total.

► [Ver en github](#)

Repaso general I

- Algoritmos
 - ¿Qué es un algoritmo?
 - ¿Cómo está compuesto?
 - ¿Qué formas hay de representarlo?
 - ¿Cómo probamos un algoritmo?
- Estructura de un programa en C
 - ¿Cuales son las etapas de un programa en C?
 - ¿Cual es la entrada y la salida de cada una de ellas?
- Tipos de datos en C
 - ¿Por qué debe indicarse el tipo de dato de una variable?
 - ¿Qué tipos de datos existe en C?
 - ¿Cual es el identificador de formato asociado a cada tipo de dato?
 - ¿Cómo ingreso datos a un programa en C desde el teclado?
 - ¿Cómo imprimo datos en la consola?

Repaso general II

- Operadores en C

- ¿Cuáles son los operadores?
- ¿Cuál es su precedencia? ¿Cómo lo resuelve C?
- ¿Qué son y para que se utilizan los operadores de asignación?

Ejemplos

- Estructura switch

- ¿Cuál es su diagrama de flujo?
- ¿Y el pseudocódigo?
- ¿Dónde aplicaría una estructura switch?
- ¿Cuál es su representación en diagrama de flujo y en pseudocódigo?
- ¿Cuál es su sintaxis?
- ¿Qué ocurre si no ponemos la sentencia break en los cases?
- ¿Qué ocurre si la variable de selección toma un valor no contemplado en los cases?

Repaso general III

- Operadores lógicos y relacionales
 - ¿Qué tipo de operaciones permite hacer un operador relacional?
 - ¿Cuál es el resultado?
 - ¿Qué tipos de operadores relacionales existen?
 - ¿Con qué tipo de datos trabaja un operador lógico?
 - ¿Cuál es el resultado de una operación lógica?
 - ¿Qué tipos de operadores lógicos existen?
- Estructura selectivas
 - ¿De qué tipo es la condición a evaluar?
 - ¿Cuál es su representación en diagrama de flujo y en pseudocódigo?
 - ¿Qué ocurre si la evaluación de la condición resulta verdadera?
 - ¿Y falsa?
 - ¿Cuál es la diferencia entre la estructura selectiva simple y doble?

- Estructura repetitiva for
 - ¿En qué casos es conveniente su utilización?
 - ¿Cuál es su representación en diagrama de flujo y pseudocódigo?
 - ¿Cuál es su sintaxis?
- Estructura repetitivas while y do-while
 - ¿En qué casos es conveniente su utilización?
 - ¿Cuál es su representación en diagrama de flujo y pseudocódigo?
 - ¿Cuál es su sintaxis?
 - ¿Cuál es la diferencia con el ciclo repetitivo for?
 - ¿Cuál es la diferencia entre el ciclo repetitivo while y do-while?

Ejercicios integradores - Primera etapa I

- 1 Diseñar y codificar un juego en C que permita al operador adivinar un número aleatorio comprendido entre 0 y 10. El operador sólo tiene tres intentos para adivinar el número. Para los casos fallidos, se debe imprimir un mensaje indicando si el número aleatorio es mayor o menor que la opción ingresada. Si luego de los tres intentos, el operador no logra adivinar, imprimir el mensaje "game over" y finalizar el programa. [▶ Ver en github](#)
- 2 Diseñar y codificar un programa en C que permita el ingreso de notas de un examen parcial de un curso de 108 alumnos. Al finalizar, el programa debe imprimir la nota máxima, mínima, el promedio general del curso, porcentaje de aprobados y porcentaje de reprobados. [▶ Ver en github](#)

Ejercicios integradores - Primera etapa II

- 3 Un estadio de fútbol cuenta con tres tipos de locaciones. Cada una de ellas tiene asociada un código, un precio y una cantidad como se muestra:

Código	Descripción	Costo	Cantidad disponible
10	Platea cubierta	\$100	1400 lugares
20	Platea descubierta	\$70	2000 lugares
30	Popular	\$50	5000 lugares

Diseñar y codificar un programa que permita a un operador vender entradas en este estadio de fútbol. El programa debe consultar el código de la butaca y si hay espacio disponible, proseguir con la venta. Caso contrario, se debe imprimir un mensaje de error y volver a comenzar desde el inicio.

Ejercicios integradores - Primera etapa III

Cuando el operador ingrese el código 0, significa que la venta ha finalizado y se debe imprimir:

- Porcentaje de ocupación del estadio
- Porcentaje de ubicación cada uno de los tipos de locaciones
- Recaudación total

► [Ver en github](#)