



APRENDERAPROGRAMAR.COM

BREAK: FORZAR SALIDA
BUCLES FOR, WHILE, DO EN
C. LOOP. STATEMENT NOT
WITHIN LOOP OR SWITCH
(CU00545F)

Sección: Cursos

Categoría: Curso básico de programación en lenguaje C desde cero

Fecha revisión: 2031

Resumen: Entrega nº45 del curso básico "Programación C desde cero".

Autor: Mario Rodríguez Rancel

SALIDA FORZADA DE BUCLES USANDO BREAK

La salida forzada de un bucle *while* se puede conseguir con el uso de la sentencia *break*; de la misma forma que hemos indicado para los bucles *for*. Cuando existe anidamiento de bucles, se produce la salida del bucle más interno al inmediato superior. Si se escribe un *break* fuera de un bucle o *switch* aparece el mensaje: "*break statement not within loop or switch*".



Ejecuta este pequeño programa y comprueba cómo estas instrucciones permiten modificar el flujo natural de un bucle *while*.

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int i=0; int mostrar=1;
    while (i<5) { if(i>3) { break; }
                printf("i vale %d\n", i);
                i++; }
    i=0; // Ejemplos aprendereaprogramar.com
    while (i<10) { if(i%3==0) {mostrar=0;}
                  if (mostrar) { printf("Ahora i vale %d\n", i); }
                  i++; mostrar = 1; }
    return 0;
}
```

El resultado de ejecución esperado es el siguiente: i vale 0, i vale 1, i vale 2, i vale 3

Ahora i vale 1, Ahora i vale 2, Ahora i vale 4, Ahora i vale 5, Ahora i vale 7, Ahora i vale 8

En el primer bucle, comprobamos cómo al verificarse la condición $i > 3$, se produce la salida forzada del bucle. En el segundo bucle, comprobamos cómo cuando el resto de dividir el valor del índice entre tres resulta cero, se omite la ejecución de un fragmento de código controlado por un condicional. En este ejemplo, no se muestran los valores 0, 3, 6, 9 porque el resto de su división por tres es cero.

FORZAR LA SALIDA DE BUCLES DO ... WHILE CON BREAK. SIMULAR BUCLES DO ... LOOP Y SALIDA

Dado que la construcción *do ... while* es análoga a *while ...*, la forma de forzar la salida de un bucle *do ... while* es la misma que ya conocemos, basada en el uso de *break*:

La construcción *Hacer ... Repetir* de la que hablamos al estudiar pseudocódigo no existe como tal en C, aunque se puede simular estableciendo una condición para un bucle *while* que siempre se evalúe a verdadero, como por ejemplo *while (1)*. Esto nos obliga, necesariamente, a que exista una condición de salida interna y a que exista un *break* dentro del bucle. Consideremos el siguiente ejemplo:

```

#include <stdio.h>
#include <stdlib.h>
// Ejemplos aprenderaprogramar.com
int main() { int i=0;
             while(1) { printf("i vale %d\n", i);
                       i = i + 1;
                       if (i > 5) {break;}
                       }
             return 0;
}

```

En caso de un fallo del contador o de la condición de salida se producirá un bucle infinito.

Es poco recomendable usar construcciones de este tipo como elemento habitual de programación. Lo más razonable es usarlas únicamente en casos excepcionales y en líneas generales, por el contrario, prescindir de ellas.

EJERCICIO RESUELTO N°1: ENUNCIADO

Transformar en código el siguiente pseudocódigo y razonar para comprender su lógica. Se trata de un ejemplo de aplicación de *SalirHacer* y *Hacer ... Repetir*, que debemos implementar en C mediante la sentencia *break*; en un bucle *while* con una condición de entrada siempre verdadera.

Nota: Considerar que el valor de *SecretCode* se encuentra en el archivo *code.dat*, creado previamente.

```

1. Inicio [Mensaje si se acierta número secreto]
2. Leer Secretcode [Valor del número secreto]
3. Hacer
    Mostrar "Introduzca número secreto": Pedir Numero
    Si Numero = Secretcode Entonces
        Mostrar "Bienvenido. Su clave para operar será 685297"
        SalirHacer
    SiNo
        Mostrar "Número erróneo": i = i + 1
        Si i > 9 Entonces
            Mostrar "Excedido número de pruebas disponibles. Acceso denegado"
            SalirHacer
        FinSi
    FinSi
Repetir
4. Fin [Pseudocódigo aprenderaprogramar.com]

```

EJERCICIO RESUELTO N°1: SOLUCIÓN

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int i=0;
    double secretCode=0.0; double numero=0.0;
    puts ("Programa codigo secreto \n");
    FILE* fichero; fichero = fopen("code.dat", "rt");
    fscanf (fichero, "%lf", &secretCode );
    fclose(fichero);
    while (1) {
        if (i > 9) {
            printf( "Excedido numero de pruebas disponibles. Acceso denegado");
            break; // Ejercicios resueltos aprenderaprogramar.com
        }
        printf("Introduzca numero secreto: "); scanf ("%lf", &numero);
        if (numero==secretCode) {
            printf ("Bienvenido. Su clave para operar sera 685297 \n");
            break;
        } else { i = i + 1;
            printf("Numero erroneo\n"); }
        }
    return 0;
}
```

EJERCICIO

Estudia el siguiente código y responde a las preguntas:

```
#include <stdio.h>
#include <stdlib.h>
// Curso de programación C aprenderaprogramar.com
int main () {
    /* local variable definition */
    int a = 10;
    /* while loop execution */
    while( a < 20 ) {
        printf("value of a: %d\n", a);
        a++;
        if( a > 15) {
            break;
        }
    }
    return 0;
}
```

- a) Sin ejecutar el código (sólo pensando) responde: ¿cuál será el resultado del programa?
- b) Ejecutando el código ¿cuál es el resultado del programa?
- c) ¿Qué significado tiene a++ en el contexto de este programa?
- d) ¿Es equivalente escribir `if(a > 15) break;` y `if(a > 15) { break; }` ?
- e) Haz que el código cuente desde 10 hasta 15 sin hacer uso de break (usa sólo las condiciones naturales de un bucle while).

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00546F

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=82&Itemid=210