



APRENDERAPROGRAMAR.COM

FINALIZAR PROGRAMAS EN
C CON EXIT. CAMBIAR
FLUJO EN BUCLES BREAK Y
CONTINUE. EJEMPLOS
(CU00544F)

Sección: Cursos

Categoría: Curso básico de programación en lenguaje C desde cero

Fecha revisión: 2031

Resumen: Entrega nº44 del curso básico "Programación C desde cero".

Autor: Mario Rodríguez Rancel

CAMBIAR EL FLUJO NATURAL DE PROGRAMAS EN C

Las recomendaciones para el uso de instrucciones de control directo del flujo de programas se resumen en una palabra: prudencia. Recomendamos leer lo expuesto al respecto en el curso "Bases de la programación nivel I" de aprenderaprogramar.com, donde se explican las pautas generales a seguir a la hora de realizar modificaciones del flujo natural de programas.



FUNCIÓN EXIT

En programación C disponemos de la función *exit* que permite finalizar de forma controlada un programa. La forma de uso habitual será:

```
exit (-1);
```

Donde -1 es un valor devuelto por la función al ambiente de ejecución (y podrá ser éste u otro; normalmente se usa -1 para indicar una detención del programa por un problema detectado).

Veamos un ejemplo de código:

```
#include <stdio.h>
#include <stdlib.h>
// Ejemplos aprenderaprogramar.com
int main() {
    int numero=0;
    printf("Introduzca un numero entero positivo: ");
    scanf("%d", &numero);
    if(numero<=0) {
        puts("Numero no valido. El programa se cierra");
        exit(-1);
    }
    printf("Gracias");
    return 0;
}
```

En este ejemplo, si el usuario introduce un número entero negativo el condicional *if* lo detecta y muestra un mensaje de advertencia mientras la función *exit* da lugar a la salida del programa.

EJERCICIO RESUELTO N°1: ENUNCIADO

Transformar en código el siguiente pseudocódigo y razonar para comprender su lógica. Se trata de un ejemplo básico de uso de la instrucción finalizar.

Pseudocódigo:

```

1. Inicio [Finaliza si hay problemas]
  2. Mientras Numero < 100 ó Numero > 200 Hacer
      Mostrar "Por favor, introduzca un número comprendido entre 100 y 200"
      Pedir Numero
      i = i + 1
      Si i > 9 Entonces
          Mostrar "Hay un problema para continuar y el programa se va a cerrar. Motivo:
          10 procesos nulos. Consulte el manual del programa"
          Finalizar
      FinSi
  Repetir
  3. Mostrar "Gracias por introducir una entrada válida"
  4. [Aquí continuaría el programa con diferentes procesos]
5. Fin [Ejercicios resueltos aprenderaprogramar.com]

```

EJERCICIO RESUELTO N°1: SOLUCIÓN

```

/* Ejemplo de uso de finalizar */
#include <stdio.h>
#include <stdlib.h>
int main() {
    int i=0; double numero = 0.0;
    printf("Ejemplo instruccion Finalizar\n\n");
    while (numero < 100 || numero > 200) {
        printf("Por favor introduzca un numero entre 100 y 200: ");
        scanf ("%lf", &numero);
        i = i + 1;
        if (i>9) {
            printf("Hay un problema para continuar y el programa se va a cerrar.\n");
            printf("Motivo: 10 procesos nulos. Consulte el manual del programa\n");
            exit(-1); // Ejercicios resueltos aprenderaprogramar.com
        }
    }
    printf("Gracias por introducir una entrada valida");
    //Aquí continuaría el programa con diferentes procesos
    return 0;
}

```

1.1.1.1.1.1.1

ALTERAR EL FLUJO EN BUCLES FOR. BREAK Y CONTINUE

La sentencia *break*; dentro de cualquier bucle (*for*, *while*, etc.) fuerza la salida del bucle de forma inmediata sin esperar a su terminación natural. Esta sentencia en bucles podemos calificarla de poco habitual por cuanto rompe el flujo natural de los programas, situación que habitualmente se trata de evitar. No obstante, su uso puede resultar de interés en algunas situaciones especiales.

Recordar que *break*; también lo usábamos en condicionales de tipo *switch*, donde *break*; hace que una vez verificado que se cumple una opción dentro del *switch*, se salga del mismo y se continúe la ejecución en la siguiente instrucción después del *switch*. Es habitual incluir un *break*; después de cada evaluación de un *switch*, aunque no es obligatorio.

En un bucle *for*, vendría siendo equivalente a la expresión *SalirDesde* que empleamos cuando estudiamos pseudocódigo. 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 sentencia *switch* aparece el mensaje de error: "*break statement not within loop or switch*".

Otra forma de manipular el flujo en un bucle *for* es mediante el uso de la instrucción *continue*;. Con esta instrucción, insertada normalmente tras evaluar una condición, se produce el incremento automático de la variable índice y la vuelta al comienzo del bucle sin terminar la ejecución del mismo.

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

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int i=0;
    for (i=0; i<5; i++) { if(i>3) {break;} //Ejemplo break
        printf("i vale %d\n", i);
    }
    for (i=0; i<10; i++) { if(i%3==0) {continue;} //Ejemplo continue
        printf("Ahora i vale %d\n", i);
    }
    return 0; // Curso programación C aprenderaprogramar.com
}
```

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 aborta la ejecución del bucle y se vuelve al comienzo del mismo incrementando la variable índice. En este ejemplo, no se muestran los valores 0, 3, 6, 9 porque el resto de su división por tres es cero.

EJERCICIO RESUELTO N°2: ENUNCIADO

Transformar en código el siguiente pseudocódigo y razonar para comprender su lógica. Se trata de un ejemplo básico de uso de la instrucción *SalirDesde*, que debemos implementar en C mediante la sentencia *break*; en un bucle *for*. Considerar que *Numerodatos* tiene un valor fijo de 35, y que los datos se extraen usando un bucle *for* desde el archivo *notas.dat* (que habremos creado previamente introduciendo 35 líneas con valores numéricos).

```

1. Inicio [Algoritmo Media curso aprenderaprogramar.com]
2. Leer Numerodatos [Establece el nº de datos a extraer]
3. Desde i = 1 hasta Numerodatos Hacer
    Leer Dato(i) [El dato esperado en fichero debe valer entre 0 y10]
    Si Dato(i) < 0 ó Dato(i) > 10 Entonces [Dato no válido]
        Mostrar "No se pudo completar la extracción de datos.
            Motivo: dato no válido. Revise archivo y vuelva a intentarlo"
        Fallolectura = Verdadero [Interruptor]
        SalirDesde
    FinSi
    Suma = Suma + Dato(i)
    Siguiente
4. Si Fallolectura = Falso Entonces
    Media = Suma / Numerodatos: Mostrar "La media es", Media
    Sino
    Mostrar "Gracias por utilizar este programa"
    FinSi
6. Fin

```

EJERCICIO RESUELTO N°2: SOLUCIÓN

```

#include <stdio.h>
#include <stdlib.h>
#define NumeroAlumnos 35
// Ejercicios resueltos programacion aprenderaprogramar.com
int main() {
    int falloLectura = 0; //0 no hay fallo lectura, 1 hay fallo lectura
    double nota[NumeroAlumnos]; //35 elementos numerados de 0 a 34
    double sumatorio = 0;
    double media = 0;
    int i=0;
    puts ("Programa Calcular Notas Medias \n");
    //Extracción de datos de archivo
    FILE* fichero; fichero = fopen("notas.dat", "rt");
    for (i=0; i<=NumeroAlumnos-1; i++) {
        fscanf (fichero, "%lf", &nota[i] );
        if (nota[i] < 0 || nota[i] > 10) { //[Dato no válido]
            printf("No se pudo completar la extraccion de datos. Motivo: \n");
            printf("dato no valido. Revise archivo y vuelva a intentarlo \n\n");
            falloLectura = 1; //[Interruptor]
            break; //Salida forzada del for;
        }
        sumatorio = sumatorio + nota[i];
    }
    fclose(fichero);
    if (falloLectura ==0) { media = sumatorio / NumeroAlumnos;
        printf ("La nota media de la clase es %lf \n", media);
    } else { printf("Gracias por utilizar este programa \n"); }
    return 0;
}

```

Para comprobar el funcionamiento del programa, lo ejecutaremos con un archivo de datos válidos y después modificaremos un dato para hacerlo no válido, con lo cual deberá mostrarse el mensaje "No se pudo completar la extracción de datos. Motivo: dato no válido. Revise archivo y vuelva a intentarlo".

EJERCICIO N° 1

Estudia el siguiente código y responde a las preguntas:

```
#include <stdio.h>
#include <stdlib.h>
// Ejercicios aprenderaprogramar.com
int main(){
    float num,average,sum;
    int i,n;
    printf("Maximum no. of inputs\n");
    scanf("%d",&n);
    for(i=1;i<=n;++i){
        printf("Enter n%d: ",i);
        scanf("%f",&num);
        if(num<0.0)
            break;
        sum=sum+num;
    }
    average=sum/(i-1);
    printf("Average=%.2f",average);
    return 0;
}
```

- ¿Para qué crees que sirve este código?
- ¿Qué ocurre si introducimos como maximum no. of inputs 8 e introducimos los números 1.5, 12.5 y -1
¿Por qué ocurre eso? ¿Qué significado tiene lo que se muestra por pantalla?
- ¿Qué significado tiene %.2f dentro de este programa?

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

EJERCICIO N° 2

Estudia el siguiente código y responde a las preguntas:

```
#include <stdio.h>
#include <stdlib.h>
// Ejercicios aprenderaprogramar.com
int main(){
    int i,num,product;
    for(i=1,product=1;i<=4;++i){
        printf("Enter num%d:",i);
        scanf("%d",&num);
        if(num==0)
            continue;
        product*=num;
    }
    printf("product=%d",product);
    return 0;
}
```

- ¿Para qué crees que sirve este código?
- ¿Qué ocurre si introducimos como números 3, 0, -5, 2 ¿Por qué ocurre eso? ¿Qué significado tiene lo que se muestra por pantalla?
- ¿Es legal escribir un if sin los corchetes { ... }?
- ¿Qué ocurre si introducimos como números 0, 0, 0, 0 ¿Por qué ocurre eso? ¿Qué significado tiene lo que se muestra por pantalla?

Para comprobar si tus respuestas son correctas puedes consultar en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

Próxima entrega: CU00545F

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=82&Itemid=210