stephenson prieto

Aprendiendo

Lenguaje C

stephenson prieto

Aprendiendo Lenguaje C

Esquema

En esta Lección

[El comienzo]

Stephenson Prieto

Aprendiendo Lenguaje C

Guía TeleTriunfador

C es un lenguaje de programación creado en 1972 por Dennis M. Ritchie, Kenneth L. Thompson, Brian Kernighan y en los Laboratorios Bell como evolución del anterior lenguaje B.
Es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado y sobre todo, se aprende rápidamente. En poco tiempo, un programador puede utilizar la totalidad del lenguaje.

Primer Programa

La mejor forma de aprender es haciendo, a continuación realizaremos nuestro primer programa en el editor de texto:

```
1 #include <stdio.h>
2 /* Primer Programa en Lenguaje C*/
3 main( )
4 {
5 printf("Venezuela ahora es de Todos\n");
6 }
```

Analizando lo que hicimos

Ahora analizaremos nuestro primer programa ”linea por linea”:
Linea 1: Permite la inclusión de la biblioteca stdio (Standard Input Output Header) para poder utilizar instrucciones como el printf.
Linea 2: Permite incluir comentarios.
Linea 3: main() Indica el lugar a partir del cual se ejecutará el programa.
Linea 4: Indica el inicio de un bloque de instrucciones.
Linea 5: La instrucción printf permite mostrar mensajes en pantalla, las lineas de instrucciones terminan con punto y coma (\n hace un salto de linea).
Linea 6: Indica el fin de un bloque de instrucciones.

Compilando

Nuestro primer programa lo guardaremos con el nombre de **Programa.C**
Para codificar nuestros ejemplos utilizaremos el Editor de Texto de nuestro Sistema Operativo GNU/Linux, y los compilaremos con el confiable **GCC**.
Para usar GCC debemos abrir un terminal (pantalla tipo consola).

Instruciones para Compilar

```
gcc programa.c -o programa
```

Instruciones para Ejecutar

```
./programa
```

Resultado en Pantalla

Si todo salió bien, luego de ejecutado el programa, se debe leer la siguiente frase:

Venezuela ahora es de todos

Si no lo haz logrado, revisa la sintaxis del programa y vuelve a compilarlo.

Si lo haz logrado entonces ¡Felicitaciones!, vas rumbo a convertirte en un Programador de la República Bolivariana de Venezuela.

Venezuela
AHORA ES DE TODOS

En esta Lección

[Controlando la Salida]

Stephenson Prieto

Aprendiendo **Lenguaje C**

Guía TeleTriunfador

En nuestro primer programa ya vimos una instrucción que utilizaremos mucho: La función **printf.**

```
5 printf("Venezuela ahora es de Todos\n");
```

La función printf

La función **printf** toma como argumento una cadena de caracteres, que se imprimen y van encerradas entre comillas dobles **" "**.
En pocas palabras, todo lo que esté dentro de las comillas, saldrá impreso por pantalla al momento de la ejecución del programa.
En nuestro ejemplo saldrá todo a excepción del símbolo **\n** el cual indica un cambio de línea.

Los Caracteres de Control

Estos sirven para controlar la salida de datos por pantalla.

\a Alerta

\n Salto de línea

\b Espacio atrás

\v Tabulación vertical

\r Retorno de carro

\t Tabulación horizontal

\f Salto de página

Programa Propuesto

Usando la instrucción **printf** y los caracteres de control, realice un código fuente en Lenguaje C que de como resultado la siguiente pantalla:

Por si acaso

Tal vez en alguna ocasión usted desee mostrar por pantalla algunas comillas o barras. Para no causar conflictos con el compilador debe usar los caracteres de control de la siguiente manera:

\' Comilla simple

**** Barra invertida

\" Comillas dobles

Dicho esto continuemos a paso firme y ***triunfadores como en Ayacucho.***

MISIÓN SUCRE

En esta Lección

[Tipos de Datos]

Stephenson Prieto

Aprendiendo **Lenguaje C**

Guía TeleTriunfador

A toda variable que se use en un programa, se le debe asociar (generalmente al principio del programa) un tipo de dato específico.

Tipos de Datos

Un tipo de dato define todo el posible rango de valores que una variable puede tomar al momento de ejecución del programa y a lo largo de toda la vida útil del propio programa.
Entre los mas comunes tenemos los siguientes:

int Tipo de Dato Entero

long Tipo de Dato Entero Largo

float Tipo de Dato Real

double Tipo de DatoReal de doble precisión

char Tipo de Dato caracter

Tambien existen arreglos y cadenas de caracteres

Formato de Escritura

Estos sirven para controlar el formato de salida de datos.

%u Imprime Entero sin signo
%d Imprime Entero
%i Imprime Entero
%ld Imprime Entero Largo
%f Imprime Real
%lf Imprime real de Doble Precisión
%e Imprime en forma Exponencial
%g Imprime %f o %e en función del tamaño del número
%c Imprime Caracter
%s Imprime una cadena de caracteres

Programa de Ejemplo

```
1  #include <stdio.h>
2  /* Trabajando con Datos*/
3  int x;
4  float y;
5  main( )
6  {
7  x= 90;
8  y=35.7;
9  printf ( "Este es un valor de tipo entero %d \n",x);
10 printf ( "Este es un valor de tipo flotante %f \n",y);
11 }
```

Analizando lo que hicimos

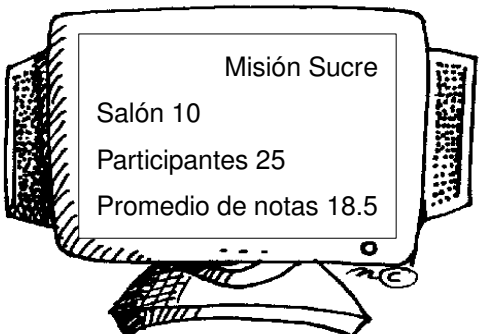
Linea 3: Declaramos x de tipo entero.
Linea 4: Declaramos y de tipo real.
Linea 7: le asignamos a x un valor entero (90).
Linea 8: le asignamos a y un valor real (35.7).
Linea 9: Imprimimos el valor de x.
Linea 10: Imprimimos el valor de y.

También puedes controlar el número de cifras en la salida, por ejemplo %2.2f muestra 2 en teros y dos decimales

Programa Propuesto

Realice un programa que tenga tres variables (mumpart, salon y prom), el valor de la primera variable será 25, el de la segunda 10 y el de la tercera 18.5

La salida por pantalla debe ser parecida a la siguiente:



En esta Lección

[Capturando Datos]

Stephenson Prieto

Aprendiendo Lenguaje C

Guía TeleTriunfador

Los datos que procesará una computadora se deben almacenar en espacios de memoria para utilizarlos posteriormente, a estos espacios se le asigna un nombre para reconocerlos, un identificador.

Identificadores

Los identificadores se forman por medio de letras, dígitos y el caracter subrayado (_). Siempre debe comenzar con letras y no debe coincidir con el listado de palabras reservadas del lenguaje. En el caso de Lenguaje C también distingue entre mayúsculas y minúsculas

Variables y Constantes

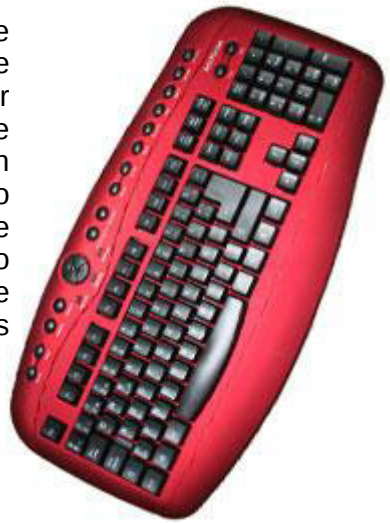
Las **variables** son objetos que pueden cambiar su valor durante la ejecución del programa, caso contrario el de las **constantes** cuyos datos no cambian durante la ejecución del programa. Para nombrarlos se utilizan identificadores.

La función scanf

La función **scanf** de la biblioteca estándar del lenguaje C permite asignar a una o más variables, uno o más valores (datos) recibidos desde la entrada estándar (el teclado).

En la fución **scanf,** por cada argumento (variable) que se le pase, se debe escribir un especificador de formato, que establecerá el formato de entrada por teclado del argumento. La forma más sencilla de escribir un especificador de formato en la función scanf es con el carácter tanto por ciento (%) seguido de un carácter de conversión que indique el tipo de dato del argumento. De momento, no es necesario conocer todos los especificadores de formato que se pueden usar en la función scanf, pero sí, los más usados, que son los mismos que en la función printf:

%d Imprime Entero
%f Imprime Real
%c Imprime Caracter
%s Imprime una cadena de caracteres



Programa de Ejemplo

```
1 #include <stdio.h>
2 /* El Duplicador:
3 Programa que duplica todo valor introcido por teclado*/
4 const int mult=2;
5 int numero;
6 int doble;
7 main( )
8 {
9 printf ( "Introduzca un número entero: ");
10 scanf ( "%d",&numero);
11 doble=numero*mult;
12 printf ( "El valor duplicado es: %d\n",doble);
13 }
```

Analizando lo que hicimos

Linea 4: Declaramos la constante ***mult*** asignándole el valor entero **2**.
Linea 5: Declaramos la variable ***numero*** de tipo entero.
Linea 6: Declaramos la variable ***doble*** de tipo entero.
Linea 10: Capturamos un valor para asignarlo a la variable ***numero***. Es importante el carácter ampersand (&) que actúa como operador de dirección.
Linea 11: Realizamos una operación matemática, multiplicando la variable ***numero*** por la constante ***mult***, el resultado lo almacenamos en la variable ***doble***.
Linea 12: Imprimimos el valor de la variable ***doble***.

Programa Propuesto

Modifique el anterior programa para que adicionalmente a su funcionalidad imprima el triple de la cifra ingresada por teclado.
Utilice una constante llamada ***multri*** y una variable llamada ***triple***.

En esta Lección

[Operaciones Aritméticas]

Stephenson Prieto

Aprendiendo Lenguaje C

Guía TeleTriunfador

C es un lenguaje donde las operaciones matemáticas pueden ser tratadas de forma fácil y eficiente gracias a la versatilidad de los operadores aritméticos y una batería de funciones especificas.

Operadores aritméticos

Los operadores aritméticos nos permiten realizar operaciones entre operandos (números, constantes y variables).

+	**Suma**	*****	**Multiplicación**	**%**	**Módulo (residuo)**
-	**Resta**	**/**	**División**		

Programa de Ejemplo

```
1 #include <stdio.h>
2 /* Operaciones Matemáticas básicas*/
3 int num1,num2;
4 int suma,resta,multi;
5 float divi;
6 main( )
7 {
8 printf ( "Introduzca un número: ");
9 scanf ( "%d",&num1);
10 printf ( "Introduzca    otro número: ");
11 scanf ( "%d",&num2);
12 suma=num1+num2;
13 resta=num1-num2;
14 multi=num1*num2;
15 divi=num1/num2;
16 printf ( "La suma es: %d\n",suma);
17 printf ( "La resta es: %d\n",resta);
18 printf ( "La multiplicación es: %d\n",multi);
19 printf ( "La división es: %lf\n",divi);
20 }
```

Analizando lo que hicimos

Linea 3, 4 y 5: Declaramos las variables necesarias.
Linea 9 y 11: Capturamos por teclado los dos números.
Linea 12, 13, 14 y 15: Realizamos las operaciones matemática.
Linea 16, 17, 8 y 19: Mostramos los resultados.

Incremento y decremento

Lenguaje C también cuenta con unos operadores propios de incremento (++) y decremento(--). Estos operadores se idearon para simplificar la escritura de fórmulas.

Programa Propuesto

Un triunfador de Misión Sucre culmina el Trayecto Inicial en la Aldea Universitaria de su municipio.
Durante un trimestre cursó cuatro materias:
•*Matemáticas I.*
•*Lenguaje y Comunicación.*
•*Proyecto Nacional y Nueva Ciudadanía*
•*Alfabetización Tecnológica.*

El triunfador posee las notas de cada materia, pero necesita que su vocero de aula le diga cual es su promedio.

Realice un programa en Lenguaje C que pregunte las notas de cada una de las materias vistas por el triunfador durante el Trayecto Inicial y calcule el promedio de notas del trimestre.

Observación: Note que una vez realizado este programa servirá para este triunfador y para cualquier otro que haya cursado el Trayecto Inicial.

En esta Lección

[Selecciones Simples]

Stephenson Prieto

Aprendiendo Lenguaje C

Guía TeleTriunfador

Lenguaje C permite la toma de decisiones mediante estructuras algorítmicas selectivas

Estructura selectiva simple *if*

La estructura selectiva ***if*** se utiliza cuando se debe tomar una decisión en el desarrollo del programa.
La toma de decisión se basa en la evaluación de una o mas condiciones que señalaran como consecuencia la rama a seguir.

Operadores relacionales

Los operadores relacionales se utilizan para comparar dos operandos, que pueden ser números, caracteres, cadenas de caracteres, constantes o variables.

==	**Igual a**
!=	**Diferente de**
<	**Menor que**
>	**Mayor que**
<=	**Menor o igual que**
>=	**Mayor o igual que**

Programa de Ejemplo

```
1 #include <stdio.h>
2 /* Programa que determina si una persona
3    es Mayor de Edad */
4 int edad;
5 main( )
6 {
7 printf ( "Introduzca su edad: ");
8 scanf ( "%d",&edad);
9 if (edad>=18)
10   {
11   printf ( "usted es mayor de edad \n");
12   }
13 }
```

Analizando lo que hicimos

Linea 9: Comparamos si la variable ***edad*** es mayor o igual a ***18.***
Linea 10: Abrimos el bloque de instrucciones de la estructura ***if.***
Linea 11: Mensaje que se mostrará si la comparación es verdadera..
Linea 12: Cerramos el bloque de instrucciones de la estructura ***if.***

Programa Propuesto

Una chica desea saber si votará en las elecciones, para eso debes realizar un programa que pregunte la edad. Si es mayor de edad imprimirá un mensaje afirmativo, si es menor de edad imprimirá un mensaje negativo (debes usar dos estructuras selectivas).

El valor referencial de la mayoría de edad debe estar en una constante declarada al comienzo del programa.

CNE PODER ELECTORAL La Democracia Evoluciona

En esta Lección

[Selecciones dobles]

Stephenson Prieto

Aprendiendo **Lenguaje C**

Guía TeleTriunfador

Lenguaje C permite la toma de decisiones mediante estructuras algorítmicas selectivas

Estructura selectiva doble *if-else*

La estructura selectiva doble ***if-else*** **permite la bifurcación del programa en dos ramas.**
Si al evaluar la condición el resultado es verdadero, se sigue por un camino especifico; si el resultado es falso, entonces sigue por otro camino predefinido.

Programa de Ejemplo

```
1  #include <stdio.h>
2  /* Programa que determina si un estudiante
3     está aprobado o reprobado */
4  int nota;
5  main( )
6  {
7  printf ( "Introduzca su nota: ");
8  scanf ( "%d",&nota);
9  if (nota>=10)
10    {
11    printf ( "Aprobado \n");
12    }
13 else
14    {
15    printf ( "Reprobado \n");
16    }
17 }
```

Analizando lo que hicimos

Linea 9: Comparamos si la variable ***nota*** es mayor o igual a ***10***.
Linea 11: Mensaje que se mostrará si la comparación es verdadera.
Linea 11: ***else***, sígnica "sino".
Linea 15: Mensaje que se mostrará si la comparación es falsa.

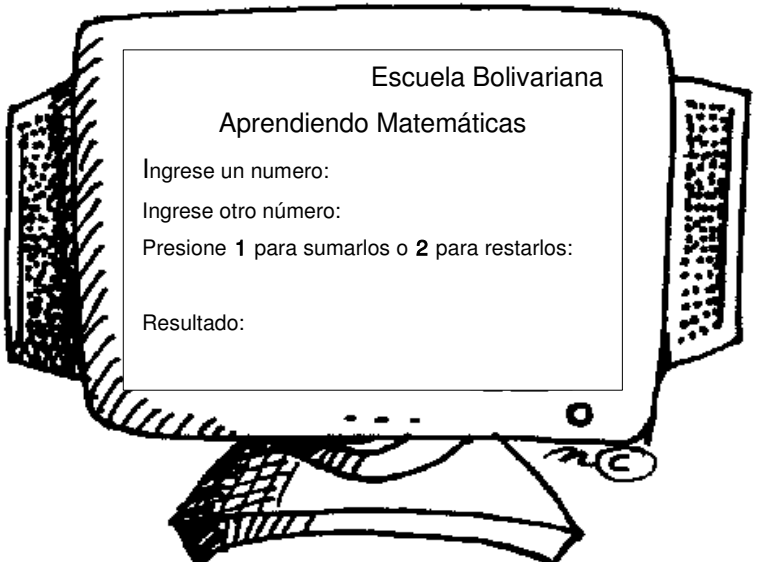
Programa Propuesto

Una Escuela Bolivariana está diseñando un software educativo llamado *"Aprendiendo Matemáticas"*.

El Ministerio de Educación requiere de su ayuda para programar uno de los módulos en Lenguaje C.

Ministerio del Poder Popular para la Educación

El programa debe preguntar dos números para ser ingresados por teclado. Posteriormente el estudiante debe teclear el ***numero 1*** si desea sumarlos o el ***número 2*** si desea restarlos.
La ejecución del programa debe ser parecida a la siguiente pantalla:



Para la programación de este módulo debe utilizar una estructura selectiva doble ***if-else.***

En esta Lección

[Condiciones Lógicas]

Stephenson Prieto

Aprendiendo Lenguaje C

Guía TeleTriunfador

Los operadores lógicos nos permiten crear condicionantes mas complejas

Operadores lógicos

&&	**Conjunción**	(el equivalente de la palabra **Y** en pseudocódigo o AND de otros lenguajes)
II	**Disyunción**	(el equivalente de la palabra **O** en pseudocódigo u **OR** de otros lenguajes)
!	**Negación**	

Programa de Ejemplo

```
 1 #include <stdio.h>
 2 /* Programa que determina si un estudiante
 3    es sobresaliente */
 4 int nota;
 5 main( )
 6 {
 7 printf ( "Introduzca su nota: ");
 8 scanf ( "%d",&nota);
 9 if ( (nota>=18) && (nota<=20) )
10    {
11    printf ( "Alumno Sobresaliente \n");
12    }
13 }
```

Analizando lo que hicimos

Linea 9: Comparamos si la variable ***nota*** es mayor o igual a ***18*** Y que a su vez sea menor o igual a 20.
Linea 11: Mensaje que se mostrará si la comparación es verdadera.

El programa sólo debería mostrar el mensaje con los valores 18, 19 ó 20.

Cuando usar cada una

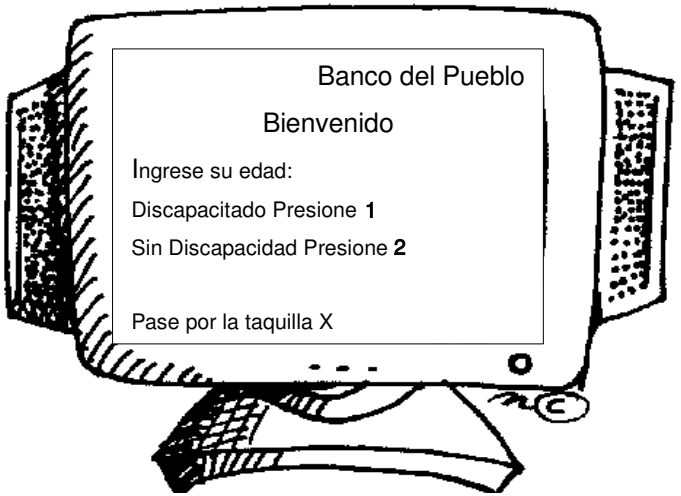
La conjunción (&&) se utilizará cuando se requiera que las dos condicionantes sean ciertas.
La disyunción (**II**) se utilizará cuando basta conque sólo una de las dos condicionantes sean ciertas.

Programa Propuesto

El **Banco del Pueblo** necesita programar un dispensador de boletos electrónicos para esperar turno en el banco.

El código fuente del software que manejará el equipo estará codificado en Lenguaje C.

Banco del Pueblo Soberano



Se requiere que el computador pregunte la edad del cliente y seguidamente pregunte si posee alguna discacidad física.

Si el cliente tiene mas de 60 años o posee alguna discapacidad el computador debe direccionarlo a la Taquilla número 1, en caso contrario lo direccionará a cualquier otra taquilla.

En esta Lección

[Selecciones Múltiples]

Stephenson Prieto

Aprendiendo Lenguaje C

Guía TeleTriunfador

Lenguaje C permite la toma de decisiones no sólo mediante estructuras algorítmicas selectivas simples, sino también múltiples

Estructura selectiva múltiple *switch*

La estructura selectiva múltiple switch permite que el flujo del diagrama se bifurque por varias ramas en el punto de la toma de decisión. La elección del camino a seguir depende del contenido de la variable conocida como selector, la cual puede tomar valores de un conjunto previamente establecido. El camino elegido, entonces, dependerá del valor que tome el selector.

Programa de Ejemplo

```
1 #include <stdio.h>
2 /* Programa de Selección Múltiple*/
3 int x,y,op;
4 float result;
5 main( )
6 {
7 printf ("\n\n\n");
8 printf ("Ingrese un número: \t");
9 scanf ("%d",&x);
10 printf ("Ingrese otro número: \t");
11 scanf ("%d",&y);
12 printf ("\n 1-Suma\n 2-Resta\n 3-Multiplicación\n 4-División\n");
13 printf ("¿Operación a realizar?: \t");
14 scanf ("%d",&op);
15 switch (op)
16         {
17         case 1: result=x+y;break;
18         case 2: result=x-y;break;
19         case 3: result=x*y;break;
20         case 4: result=x/y;break;
21         }
22 printf ("\n\n\n");
23 printf ("Eligió la opción %d cuyo resultado es: %2.2f",op,result);
24 printf ("\n\n\n");
25 }
```

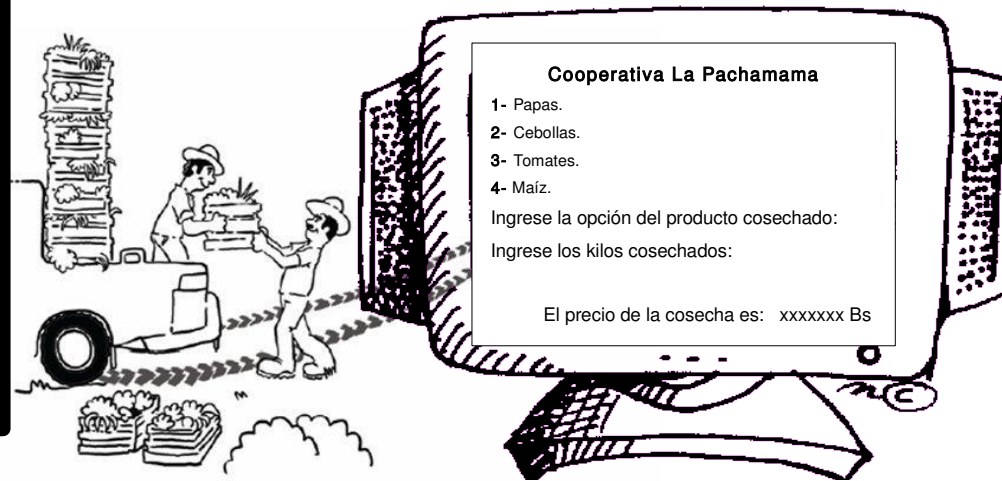
Analizando lo que hicimos

Linea 9 y 11: Capturamos los números en las variables ***x*** y ***y.***
Linea 12: Imprimimos el menú de opciones en pantalla*.*
Linea 14: Capturamos la variable ***op***, que fungirá de selector.
Linea 17: Suma, en caso de elegir la opción **1.**
Linea 18: Resta, en caso de elegir la opción **2.**
Linea 19: Multiplicación, en caso de elegir la opción **3.**
Linea 20: División, en caso de elegir la opción **4.**
Linea 23: Se imprime el resultado en pantalla.

Programa Propuesto

Una cooperativa de agricultores realizan unas cosechas de papas, cebollas, tomates y maíz.
El valor de cada kilo es papas 5.000Bs, cebollas 9.000Bs, tomates 8.000Bs y maíz 7.000Bs.

Debido a un crédito especial para automatizar sus procesos, los agricultores han adquirido unas computadoras pero necesitan un programa específico que calcule la cantidad de dinero que deben pedir por su cosecha.



En esta Lección

[Ciclos repetitivos definidos]

Stephenson Prieto

Aprendiendo **Lenguaje C**

Guía TeleTriunfador

Durante la solución de Problemas, es muy común encontrar operaciones que deben ejecutarse un determinado número de veces.

Estructura repetitiva for

Esta estructura algorítmica se utiliza para repetir un conjunto de instrucciones un número definido de veces.

Programa de Ejemplo

```
1  #include <stdio.h>
2  /* Programa de Estrucutras Ciclicas Determinadas*/
3  int x,n,a;
4  main( )
5  {
6  for (x=1;x<=5;x++)
7          {
8          printf ("Ingrese un número: \t");
9          scanf ("%d",&n);
10         a=a+n;
11         }
12 printf ("\nEl resultado de la suma de los números es: %d\n",a);
13 }
```

Analizando lo que hicimos

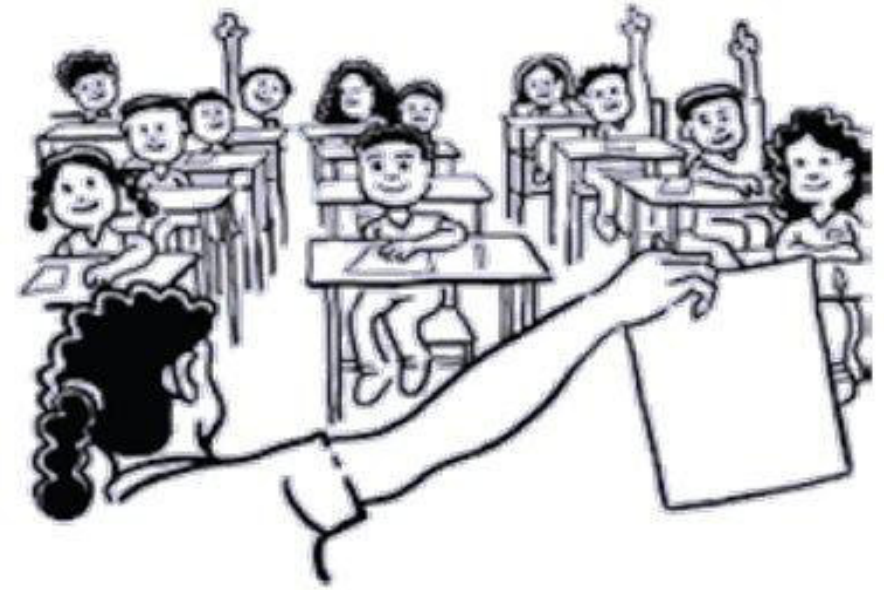
Linea 6: utilizamos un ***for***, donde la variable ***x*** se inicializará en 1, llegará hasta 5, en forma creciente de uno en uno.
Linea 9: Se captua la variable ***n***, esta acción se repetirá 5 veces.
Linea 10: Se utiliza la variable ***a*** como acumulador, esta irá acumulando los valores que toma ***n*** y los sumará.
Linea 12: Se imprime el resultado en pantalla.

Programa Propuesto

Realice un programa codificado en lenguaje C, donde se pregunte a 15 alumnos sus edades, posteriormente se debe mostrar en pantalla el promedio de edad de los alumnos del salón.

Ministerio del Poder Popular para la **Educación**

Este programa se ejecutará en una Escuela Bolivariana. Recuerde utilizar para este programa la estrucutura repetitiva for.



Extra

Usando lo aprendido en lecciones pasadas, agréguele las siguientes funcionalidades al programa:
•Si el promedio de edad es menor a 13 años, emitir un mensaje en pantalla que diga que la población es de niños y niñas, en caso contrario el mensaje dirá que son adolescentes.

En esta Lección

[Ciclos repetitivos indefinidos]

Stephenson Prieto

Aprendiendo **Lenguaje C**

Guía TeleTriunfador

Durante la solución de Problemas, es muy común encontrar operaciones que deben ejecutarse un determinado número de veces.

Estructura repetitiva while

Esta estructura algorítmica se utiliza permite repetir un conjunto de instrucciones. Sin embargo el número de veces que debe repetirse depende de las proposiciones que tenga el ciclo. Cada vez que corresponde iniciar el ciclo se evalúa una condición, si ésta es verdadera (diferente de cero) se continúa con la ejecución, de otra forma se detiene.

Programa de Ejemplo

```
1  #include <stdio.h>
2  /* Programa de Estrucutras Ciclicas Indeterminadas (while)*/
3  /* Este Programa suma números hasta que se le ingrese un cero*/
4  int n,a;
5  main( )
6  {
7  printf ("Ingrese un número: \t");
8  scanf ("%d",&n);
9  while (n!=0)
10         {
11         a=a+n;
12         printf ("Ingrese otro número: \t");
13         scanf ("%d",&n);
14         }
15 printf ("\nEl resultado de la suma de los números es: %d\n",a);
16 }
```

Analizando lo que hicimos

Linea 8: Capturamos el valor de un número.
Linea 9: Utilizando la estructura ***while***, evaluamos si el número anteriormente capturado es diferente de cero.
Linea 11: Se utiliza la variable ***a*** como acumulador, esta irá acumulando los valores que toma ***n*** y los sumará.
Linea 8: Volvemos a capturar el valor de un nuevo número.
Linea 12: Se imprime el resultado en pantalla.

Programa Propuesto

Un establecimiento de Mercal tiene una cantidad limitada de pollos para venderlo a la comunidad, sólo se venderá el producto hasta agotarse la existencia.
Diseñe un programa codificado en Lenguaje C que pregunte si hay pollos en existencia, en caso de ser afirmativo se debe vender una unidad.

Para efectos del programa, al momento de preguntar la existencia de pollos, un 1 indicará SI y un 2 indicará NO.

Al final debe salir un mensaje que diga “Existencia Agotada”

Extra

Usando lo aprendido en lecciones pasadas, agréguele las siguientes funcionalidades al programa:
•Número de pollos vendidos.

En esta Lección

[Ciclos repetitivos indefinidos]

Stephenson Prieto

Aprendiendo Lenguaje C

Guía TeleTriunfador

Durante la solución de Problemas, es muy común encontrar operaciones que deben ejecutarse un determinado número de veces.

Estructura repetitiva do - while

Esta estructura algorítmica se utiliza permite repetir un conjunto de instrucciones. A diferencia de las estructuras *for* y while, en las cuales las condiciones se evalúan al principio del ciclo, en ésta se evalúan al final. Esto implica que el ciclo se debe ejecutar por lo menos una vez.

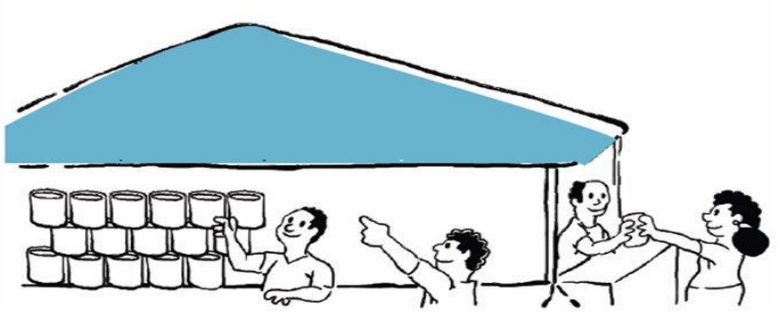
Programa de Ejemplo

```
1  #include <stdio.h>
2  /* Programa de Estrucutras Ciclicas Indeterminadas (do-while)*/
3  /* Este Programa suma números hasta que el resultado sea mayor a 100*/
4  int n,a;
5  main( )
6  {
7  do
8          {
9          printf ("Ingrese un número: \t");
10         scanf ("%d",&n);
11         a=a+n;
12         }
13 while (a<100);
14 printf ("\nEl resultado de la suma de los números es: %d\n",a);
15 }
```

Analizando lo que hicimos

Linea 7: Comienza el do.
Linea 10: Capturamos el valor de un numero.
Linea 11: Se utiliza la variable ***a*** como acumulador, esta irá acumulando los valores que toma ***n*** y los sumará.
Linea 13: Con ***while*** evaluamos si el acumulador es menor a cero. Si es cierto, se repite el ciclo, en caso contrario se detiene.
Linea 14: Se imprime el resultado en pantalla.

Programa Propuesto



Un Mercal itinerante se establece en una populosa barriada. Tiene una cantidad ilimitada de potes de leche en polvo para venderlo a la comunidad, sólo se venderá el producto hasta que no quede gente haciendo cola.
Diseñe un programa codificado en Lenguaje C que pregunte si hay gente en cola, en caso de ser afirmativo se debe vender un pote. Para efectos del programa, al momento de preguntar la existencia de potes, un 1 indicará SI y un 2 indicará NO.

Extra

Usando lo aprendido en lecciones pasadas, agréguele las siguientes funcionalidades al programa:
•Número de potes vendidos.
•Cantidad de dinero recogido (a razón de 16Bs el pote).

Apéndices

stephenson prieto

Aprendiendo

Lenguaje C

Apéndice 1

Planilla de Evaluación

	Lección 1		Lección 2		Lección 3		Lección 4		Lección 5		Lección 6		Lección 7		Lección 8		Lección 9		Lección 10		Lección 11		Lección 12	
	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto	Ejemplo	Propuesto

En cada clase (a excepción de la 1era y la segunda) el ejercicio de ejemplo vale 05 puntos y el ejercicio propuesto vale 15, el estudiante que en cada clase hace los dos ejercicios correctamente tiene 20 puntos.
Al final del trimestre se promediaran las notas de todas las clases.

Las tres últimas lecciones tienen un apartado de extra, esto permitirá agregar puntos a juicio del profesor.

Apéndice 2

Librerías ANSI C

El estándar ANSI C define un conjunto de funciones, así como tipos relacionados y macros, que son proporcionados para la implementación. Todas las librerías son declaradas en un fichero cabecera. Para que sea visible al programa, se añade el comando del preprocesador #include. Por ejemplo: #include <stdio.h>;

Cada fichero de cabecera se denomina librería. En la siguiente lista mostraremos la Librería junto con la descripción:

•**assert.h** Contiene una macro para el diagnóstico dentro de los programas.
•**ctype.h** Contiene varias funciones para comprobación de tipos y transformación de caracteres.
•**errno.h** Contiene varias macros usadas para informar de errores.
•**limits.h** Contienen varias macros que definen constantes para el tamaño de tipo enteros.
•**float.h** Contienen varias macros que definen constantes para el tamaño de tipo flotante.
•**locale.h** Contienen varias macros, funciones y tipos para unidades locales, como unidad monetaria, tiempo, dígitos, etc.
•**math.h** Contiene una macro y varias funciones matemáticas.
•**setjmp.h** Contienen declaraciones que proporcionan una forma de evitar la secuencia normal de llamada y regreso de funciones.
•**signal.h** Contiene un tipo, dos funciones y varias macros para manejar condiciones excepcionales que aparecen durante la ejecución, tal como una señal de interrupción de una fuente externa o un error en la ejecución.
•**stdarg.h** Contiene un tipo y tres macros que proporcionan recursos para recorrer una lista de argumentos de función de tamaño y tipo desconocido.
•**stddef.h** Contiene varios tipos y macros que también están definidas en otras librerías, como size_t.
•**stdio.h** Contiene tipos, macros y funciones para la realización de tareas de E/S.
•**stdlib.h** Contiene tipos, macros y funciones para la conversión numérica, generación de números aleatorios, búsquedas y ordenación, gestión de memoria y tareas similares.
•**string.h** Contiene tipos, macros y funciones para la manipulación de cadenas de caracteres.
•**time.h** Contiene tipos, macros y funciones para la la manipulación de información sobre fechas y horas.

Editorial

TeleTriunfador