



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 8

Integrante(s): Ramírez Pérez Daniela Itzel

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: 36

Semestre: 1er

Fecha de entrega: Viernes 4 de Diciembre

Observaciones:

CALIFICACIÓN: _____

Objetivo

Elaborar programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria (o condicional) para la resolución de problemas básicos.

Introducción

Las estructuras de control de flujo en un lenguaje especifican el orden en que se realiza el procesamiento de datos.

Los lenguajes de estructuras de selección:

- estructura if-else
- estructura switch
- estructura condicional o ternaria.

Estructura de control selectiva if

La estructura de control de flujo más simple es la estructura condicional if, su sintaxis es la siguiente:

```
if (expresión_lógica) {  
    // bloque de código a  
    ejecutar  
}
```

En esta estructura se evalúa la expresión lógica y, si la condición es verdadera, se ejecutan las instrucciones del bloque que se encuentra entre las llaves de la estructura.

Si no se cumple la condición, se continúa con el flujo normal del programa.

Código (estructura de control selectiva if)

```
#include<stdio.h>
/*
Este programa valida si el número a es mayor al número b.
*/
int main (){
    int a, b;
    a = 3;
    b = 2;

    if (a > b) {
        printf("\ta (%d) es mayor a b (%d).\n",a,b);
    }

    printf("\t\vEl programa sigue su flujo.\n");
    return 0;
}
```

a (3) es mayor a b (2).
El programa sigue su flujo.

Código (estructura de control selectiva if)

```
#include<stdio.h>
/*
Este programa comprueba que las condiciones son numéricas
0 -> falso
≠ 0 -> Verdadero
*/
int main(){
    char sp=168,si=63,ae=130,ai=161,au=163,ao=162,aa=160;
    if (0){
        printf("Esta instrucciócn nunca se ejecuta\n",ao);
        printf("porque la condiciócn siempre es falsa (0).\n",ao);
    }

    if (-38)
        // El bloque de código de esta estructura if
        // solo consta de una línea porque los comentarios
        // no son tomados en cuenta por el compilador.
        // La condición siempre es verdadera (diferente de 0)
        printf("Esta instrucciócn siempre se ejecuta.\n",ao);

    return 0;
}
```

Esta instrucción siempre se ejecuta.

Estructura de control selectiva if-else

```
if (expresión_lógica) {  
    // bloque de código a ejecutar  
    // si la condición es verdadera  
} else {  
    // bloque de código a ejecutar  
    // si la condición es falsa  
}
```

Esta estructura evalúa la expresión lógica y si la condición es verdadera se ejecutan las instrucciones del bloque que se encuentra entre las primeras llaves, si la condición es falsa se ejecuta el bloque de código que está entre las llaves después de la palabra reservada 'else'.

Al se continúa con el flujo normal del programa.

Estructura de control selectiva switch-case

La sintaxis de la estructura switch-case es la siguiente:

```
switch (opcion_a_evaluar){  
    case valor1:  
        /* Código a ejecutar*/  
        break;  
    case valor2:  
        /* Código a ejecutar*/  
        break;  
    ...  
    case valorN:  
        /* Código a ejecutar*/  
        break;  
    default:  
        /* Código a ejecutar*/  
}
```

La estructura switch-case evalúa la variable que se encuentra entre paréntesis después de la palabra reservada switch y la compara con los valores constantes que posee cada caso (case).

Los tipos de datos que puede evaluar esta estructura son enteros, caracteres y enumeraciones. Al final de cada caso se ejecuta la instrucción break, si se omite esta palabra reservada se ejecutaría el siguiente caso, es decir, se utiliza para indicar que el bloque de código a ejecutar ya terminó y poder así salir de la estructura.

Si la opción a evaluar no coincide dentro de algún caso, entonces se ejecuta el bloque por defecto (default). El bloque por defecto normalmente se escribe al final de la estructura, pero se puede escribir en cualquier otra parte. Si se escribe en alguna otra parte el bloque debe terminar con la palabra reservada break.

Enumeración

Existe otro tipo de dato constante conocido como enumeración. Una variable enumeradora se puede crear de la siguiente manera:

```
enum identificador {VALOR1, VALOR2, ... , VALORN};
```

Para crear una enumeración se utiliza la palabra reservada enum, seguida de un identificador (nombre) y, entre llaves se ingresan los nombres de los valores que puede tomar dicha enumeración, separando los valores por coma. Los valores son elementos enteros y constantes (por lo tanto se escriben con mayúsculas).

Ejemplo

```
enum boolean {FALSE, TRUE};
```

La enumeración se llama 'boolean' y contiene dos elementos, el primero (FALSE) posee el valor 0 y el siguiente (TRUE) posee el valor 1. Si hubiese más elementos en la enumeración, la numeración correría de manera ascendente.

Es posible cambiar el valor de un elemento, para ello solo se le asigna el valor deseado:

```
enum diasSemana {LUNES, MARTES, MIERCOLES=5, JUEVES, VIERNES};
```

Código (variables tipo enumeración)

```
#include <stdio.h>
/*
 Este programa crea diversas variables tipo enum (enumerador) y
 permite visualizar la manera en la que se maneja el tipo de dato.
 */
int main(){
 // declaración de la enumeración
 enum boolean {NO, YES};

 // declaración de una variable tipo enumeración
 enum boolean valorBooleano;
 valorBooleano = YES;

 // Se comprueba que el valor de una enumeración es entero
 printf("%d\n", valorBooleano);

 // Se comprueba que el valor de una enumeración se puede reasignar
 enum diasSemana {LUNES, MARTES, MIERCOLES=5, JUEVES, VIERNES};
 printf("\n%d", LUNES);
 printf("\n%i", MARTES);
 printf("\n%d", MIERCOLES);
 printf("\n%i", JUEVES);
 printf("\n%d\n", VIERNES);

 return 0;
}
1
0
1
5
6
7
```

Estructura de control selectiva condicional

La estructura condicional (también llamado operador ternario) permite realizar una comparación rápida. Su sintaxis es la siguiente:

Condición ? SiSeCumple : SiNoSeCumple

Consta de tres partes, una condición y dos acciones a seguir con base en la expresión condicional. Si la condición se cumple (es verdadera) se ejecuta la instrucción que se encuentra después del símbolo '?'; si la condición no se cumple (es falsa) se ejecuta la instrucción que se encuentra después del símbolo ':'.

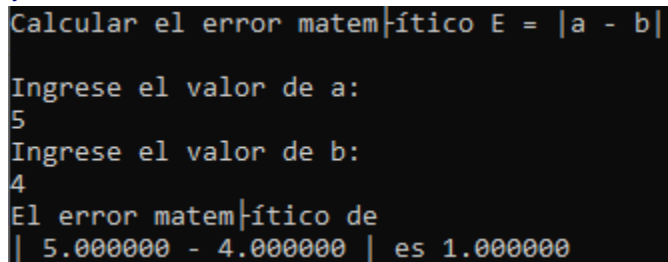
Código (Estructura de control selectiva condicional o ternaria)

```
#include <stdio.h>
/*
Este programa permite calcular el error matemático a partir de dos
valores (a y b) ingresados desde la entrada estándar (el teclado), a partir
de la fórmula:
 $E = |a - b|$ 
Donde a es el valor real y b es el valor aproximado o viceversa.
*/
int main(){
    double a, b, res;
    printf("Calcular el error matemático E = |a - b|\n\n");
    printf("Ingrese el valor de a:\n");
    scanf("%lf",&a);
    printf("Ingrese el valor de b:\n");
    scanf("%lf",&b);

    res = a < b ? b-a : a-b;

    printf("El error matemático de\n");
    printf("| %lf - %lf | es %lf\n", a, b, res);

    return 0;
}
```



```
Calcular el error matemático E = |a - b|

Ingrese el valor de a:
5
Ingrese el valor de b:
4
El error matemático de
| 5.000000 - 4.000000 | es 1.000000
```

Conclusion

Gracias a la practica se tiene un mejor entendimiento de las diferentes aplicaciones a las que tiene acceso las diferentes estructuras de control para que nuestros futuros proyectos puedan tener diferentes funciones para los problemas que se requieran resolver en su momento.

Referencias

- Nakayama Cervantes, A., Castañeda Perdomo, M., Solano Gálvez, J. A., García Cano, E. E., Sandoval Montaña, L., & Arteaga Ricci, T. I. (2018, 6 abril). Manual de prácticas del laboratorio de Fundamentos de programación (p.135-147). Laboratorio de Computación Salas A y B. http://odin.fib.unam.mx/salac/practicasFP/MADO-17_FP.pdf