



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Marco Antonio Martinez Quintana

Profesor:

Fundamentos de Programación

Asignatura:

03

Grupo:

Practica N°8

No de Práctica(s):

Torres Gracian Christian Ivan

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No Aplica

55

No. de Lista o Brigada:

Semestre N°1

Semestre:

01/11/2020

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Guía práctica de estudio 08: Estructuras de selección

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.

Actividades:

- Elaborar expresiones lógicas/condicionales utilizadas en las estructuras de selección y realizar su evaluación.
- Elaborar un programa en lenguaje C para cada estructura de selección.

Introducción

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

Las estructuras de selección (o condicionales) permiten realizar una u otra acción con base en una expresión lógica. Las acciones posibles a realizar son mutuamente excluyentes, es decir, solo se puede ejecutar una a la vez dentro de toda la estructura.

Lenguaje C posee 3 estructuras de selección: la estructura if-else, la estructura switch y la estructura condicional o ternaria.

Licencia GPL de GNU

El software presente en esta práctica es libre bajo la licencia GPL de GNU, es decir, se puede modificar y distribuir mientras se mantenga la licencia GPL.

```
/*
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 *
 * Author: Jorge A. Solano
 */
```

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 se cumple (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.

NOTA 1: Si el bloque de código a ejecutar consta de una sola línea de código no es necesario el uso de las llaves.

NOTA 2: Como ya se explicó en la práctica anterior, la expresión lógica evaluada regresará como resultado un número entero. Dentro de las estructuras de control 0 indica que la expresión lógica es falsa y cualquier número diferente de 0 indica que la expresión lógica es verdadera.

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\nEl programa sigue su flujo.\n");

    return 0;
}
```

Código (estructura de control selectiva if)

```
#include<stdio.h>

/*
    Este programa comprueba que las condiciones son numéricas
    0 -> falso
    ≠ 0 -> Verdadero
*/

int main(){

    if (0){
        printf("Esta instrucción nunca se ejecuta\n");
        printf("porque la condición siempre es falsa (0).\n");
    }

    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 instrucción siempre se ejecuta.\n");

    return 0;
}
```

Estructura de control selectiva if-else

La sintaxis de la estructura de control de flujo **if-else** es la siguiente:

```
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 final de que se ejecute uno u otro código, se continúa con el flujo normal del programa.

Es posible *anidar* varias estructuras if-else, es decir, dentro de una estructura if-else tener una o varias estructuras if-else.

Código (estructura de control selectiva if-else)

```
#include <stdio.h>

/*
    Este programa permite validar si un número es par o impar.
    El número se lee desde la entrada estándar (el teclado).
*/

int main(){
    int num;

    printf("Ingrese un número:\n");
    scanf("%d",&num);

    if ( num%2 == 0 )
        printf("El número %d es par.\n",num);
    else
        printf("El número %d es impar.\n",num);

    return 0;
}
```

Código (estructura de control selectiva if-else anidada)

```
#include <stdio.h>

/*
Este programa ordena en forma descendente tres valores enteros dados.
Los valores se leen desde la entrada estándar (el teclado).
*/

int main(){
    int uno, dos, tres;

    printf ("Ingrese 3 números separados por espacios:\n");
    scanf ("%d %d %d", &uno, &dos, &tres);

    if (uno > dos){
        if (dos > tres){
            printf("%d es mayor a %d que es mayor a %d\n", uno, dos, tres);
        }else {
            if (uno > tres) {
                printf("%d es mayor a %d que es mayor a %d\n", uno, tres, dos);
            } else {
                printf("%d es mayor a %d que es mayor a %d\n", tres, uno, dos);
            }
        }
    } else {
        if (dos > tres){
            if (tres > uno) {
                printf("%d es mayor a %d que es mayor a %d\n", dos, tres, uno);
            } else {
                printf("%d es mayor a %d que es mayor a %d\n", dos, uno, tres);
            }
        } else {
            printf("%d es mayor a %d que es mayor a %d\n", tres, dos, uno);
        }
    }

    return 0;
}
```

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.

Código (estructura de control selectiva switch-case)

```
#include <stdio.h>

/*
    Este programa permite elegir una opción del menú a partir del carácter
    ingresado. La opción se lee desde la entrada estándar (el teclado).
*/

int main(){
    char op = '\0';

    printf("\tMenú\n\n");

    printf("Elegir la opción deseada\n");
    printf("a) Ingresar\n");
    printf("b) Registrarse\n");
    printf("c) Salir\n");
    scanf("%c",&op);

    switch(op) {
        default:
            printf("Opción no válida.\n");
            break;
        case 'a':
            printf("Se seleccionó 'Ingresar'.\n");
            break;
        case 'b':
            printf("Se seleccionó 'Registrarse'.\n");
            break;
        case 'c':
            printf("Se seleccionó 'Salir'.\n");
            break;
    }

    return 0;
}
```

Código (estructura de control selectiva switch-case)

```
#include <stdio.h>

/*
    Este programa permite elegir una opción del menú a partir del entero
    ingresado. La opción se lee desde la entrada estándar (el teclado).
*/

int main(){
    int op = 0;

    printf("\tMenú\n\n");
    printf("Elegir la opción deseada\n");
    printf("1) Ingresar\n");
    printf("2) Registrarse\n");
    printf("3) Salir\n");
    scanf("%d",&op);

    switch(op) {
        case 1:
            printf("Se seleccionó 'Ingresar'\n");
            break;
        case 2:
            printf("Se seleccionó 'Registrarse'\n");
            break;
        case 3:
            printf("Se seleccionó 'Salir'\n");
            break;
        default:
            printf("Opción no válida\n");
    }

    return 0;
}
```

Enumeración

Existe otro tipo de dato constante conocido como enumeración. Una variable enumerador 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;
}
```

Código (variables tipo enumeración)

```
#include <stdio.h>

/*
Este programa permite elegir una opción del menú a partir del entero
ingresado. La opción se lee desde la entrada estándar (el teclado).
*/

int main(){
    // Los valores de una enumeración son enteros y constantes
    enum diasSemana {LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO};
    int op;
    printf("Ingrese el día de la semana.\n");
    printf("1) Lunes\n");
    printf("2) Martes\n");
    printf("3) Miércoles\n");
    printf("4) Jueves\n");
    printf("5) Viernes\n");
    printf("6) Sábado\n");
    printf("7) Domingo\n");
    scanf("%d", &op);

    switch(op-1){
        case LUNES:
        case MARTES:
            printf("Inicio de semana.\n");
            break;
        case MIERCOLES:
            printf("Mitad de semana.\n");
            break;
        case JUEVES:
            printf("¡Casi inicia el fin de semana!\n");
            break;
        case VIERNES:
        case SABADO:
            printf("¡Fin de semana!\n");
            break;
        case DOMINGO:
            printf("Día de descanso.\n");
            break;
        // No se necesita default
    }

    return 0;           // Valor entero en hexadecimal
}
```

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;
}
```

Resultados

```
1  #include<stdio.h>
2  int main()
3  {
4      //mensaje de bienvenida
5      printf("\n\n\t\t\t\t\tPrograma de edades \n\n");
6
7      //Variables a utilizar
8      int edad;
9      char a=160;
10     //solicitar edad
11     printf("\tIngresa tu edad: ");
12     scanf("%d", &edad);
13
14     //implementacion if-else
15     if(edad>=18)
16     {
17         printf("Ya est&cs Grandecito!! \n\n",a);
18     }
19     else
20     {
21         printf("Todavia estas chavo!! \n");
22     }
23
24
25     return 0;
```

```

1 #include<stdio.h>
2 int main ()
3 {
4     //Bienvenida
5     printf("\n\n\t\t\t\t\tBienvenidos a la Calculadora: \n");
6
7     //Variables a utilizar
8     int op,n1,n2,res;
9     char a=160;
10    char o=162;
11    char u=163;
12
13    //Menu
14    printf(" \t1) Suma \n \t2) Resta \n \t3) Multiplicaci%cn \n \t4) Divisi%cn \n\n",o,o);
15
16    //Opciones y Numeros
17    printf("\t\tElige el inciso para realizar:");
18    scanf("%d",&op);
19
20    printf("\t\tDame dos n%cmeros separados por un espacio: ",u);
21    scanf("%i %i",&n1,&n2);
22
23    switch(op)
24    {
25        case 1:
26            res=n1+n2;
27            printf("\t\tLa Suma de %i y %i es: %d",n1,n2,res);
28            break;
29
30        case 2:
31            res=n1-n2;
32            printf("\t\tLa Resta de %i y %i es: %d",n1,n2,res);
33            break;
34
35        case 3:
36            res=n1*n2;
37            printf("\t\tLa Multiplicaci%cn de %i y %i es: %d",o,n1,n2,res);
38            break;
39
40        case 4:
41            res=n1/n2;
42            printf("\t\tLa Divisi%cn de %i y %i es: %d",o,n1,n2,res);
43            break;
44
45        default:
46            printf("\t\tOpci%cn inv%clida, Ojo con las instrucciones!!!",o,a);
47            break;
48    }
49
50    return 0;
51 }

```

Bibliografía



El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.