



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Adrián Ulises Mercado Martínez

Asignatura: Fundamentos de Programación

Grupo: 15

No de Práctica(s): 8: Estructuras de selección en C

Integrante(s): Morales Velasco Brayan, Lagunas Parra Jaime,
Walls Chávez Luis Fernando

*No. de Equipo de
cómputo empleado:* 41, 42 y 43

No. de Lista o Brigada: Brigada 13

Semestre: 2020-I

Fecha de entrega: 16/10/19

Observaciones:

CALIFICACIÓN: _____

Introducción

Las estructuras de selección, ya vistas y observadas en prácticas anteriores, son aquellas que, en base de la evaluación de una condición, nos permiten ejecutar o saltarnos ciertas líneas de código evadiendo así la estructura secuencial del programa.

En el lenguaje C contamos con 3 tipos de estructuras:

- **If-else:** Se refiere al Si utilizado en pseudocódigo y esta evalúa la condición que se escriba después de la estructura. La estructura else es omisible, pero si se llegará a utilizar ejecutaría las líneas de código dentro de este y tiene que usarse después del cierre de las instrucciones dentro de if.
- **Ternaria:** Es un caso especial de if de una sola línea donde podemos omitir las palabras por símbolos, donde se escribe la condición al principio y las separamos de las instrucciones con un "?". Estas instrucciones se ejecutarán si la condición se cumple; sino se escribe después de esas un ":" y esas serán las instrucciones a ejecutar. Sólo funciona para instrucciones de una línea.
- **Switch:** A veces se refiere a éste como un interruptor ya que permite el manejo de un sinnúmero de casos donde se evalúa un valor, a partir del cual se ejecutarán ciertas líneas de código.

Desarrollo

1.- Iniciando la práctica el primer código compilado fue uno en el cual con la instrucción **if** se comparaban dos números y sólo si el primero era menor que el segundo se imprimían en ese orden.

```
1  #include <stdio.h>
2
3  int main(){
4      int num1, num2;
5      printf("Ingresa 2 números \n");
6      scanf("%d\n", &num1);
7      scanf("%d", &num2);
8
9      if ( num1 < num2)
10         printf("El número %d es menor que %d\n ", num1, num2);
11     return 0;
```

2. En el segundo código de igual manera comparábamos 2 números, pero ahora con ayuda de **else** podíamos imprimir estos dos números empezando por el mayor.

```
1  #include <stdio.h>
2
3  int main(){
4      int num1, num2;
5      printf("Ingresa 2 números \n");
6      scanf("%d\n", &num1);
7      scanf("%d", &num2);
8
9      if ( num1 < num2)
10         printf("El primer número %d es menor que el segundo %d\n", num1, num2);
11     else
12         printf("El segundo número %d es menor que el primero %d\n ", num2, num1);
13     return 0;
14 }
```

3. En el siguiente usamos la estructura ternaria y probamos el programa anterior con esta sintaxis.

```
1  #include <stdio.h>
2
3  int main(){
4      int num1, num2;
5      printf("Ingresa 2 números \n");
6      scanf("%d\n", &num1);
7      scanf("%d", &num2);
8
9      (num1 < num2) ?
10     printf("El primer número %d es menor que el segundo %d\n ", num1, num2)
11     :
12     printf("El segundo número %d es menor que el primero %d\n", num2, num1);
13     return 0;
14 }
```

4. En el ejercicio 4 probamos añadir una instrucción más a la estructura ternaria pero se probó que con esta adición el código no se compilaba.



Proyecto Clases(Fun) [*] ejercicio1.c [*] ejercicio2.c [*] ejercicio3.c ejercicio4.c [*] ejercicio5.c [*] ejercicio6.c ejercicio7.c [*] ejercicio8

```
1  #include <stdio.h>
2
3  int main(){
4      int num1, num2;
5      printf("Ingresa 2 números \n");
6      scanf("%d\n", &num1);
7      scanf("%d", &num2);
8
9      (num1 < num2) ?
10     printf("El primer número %d es menor que el segundo %d\n ", num1, num2)
11     :
12     printf("El segundo número %d es menor que el primero %d\n", num2, num1)
13     :
14     printf("El primer número %d es igual al segundo número %d\n", num1, num2);
15     return 0;
16 }
```

Compilador (2) Recursos Registro de Compilación Depuración Resultados Cerrar

Línea	Col.	Archivo	Mensaje
13	3	D:\Descargas\Practica 8 FDP\ejercicio4.c	In function 'main':
		D:\Descargas\Practica 8 FDP\ejercicio4.c	[Error] expected ';' before ':' token

5. En los ejercicios 5, 6 y 7 agregamos la librería stdbool para usar operadores booleanos y demostramos que si se ingresaba cualquier número la condición era verdadera, pero si se ingresaba el 0 se consideraba falsa.

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  |
4  int main(){
5  |
6      if (true)
7          printf("Esta condición siempre es verdadera\n");
8      printf("Esta instrucción no pertenece al If\n");
9      return 0;
10 }
```

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  |
4  int main(){
5  |
6      if (false)
7          printf("Esta instrucción nunca se ejecuta\n");
8      printf("Esta instrucción no pertenece al If\n");
9      return 0;
10 }
```

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  |
4  int main(){
5      int a;
6      scanf("%d",&a);
7  |
8      if (a)
9          printf("Esta instrucción se ejecuta si a es diferente de 0\n");
10     printf("Esta instrucción no pertenece al If\n");
11     return 0;
12 }
```

6. En los programas 8 y 9 ahora comparamos 3 números, pero en el 9 ocupamos la estructura else para ahorrarnos la escritura de unas cuántas líneas.

```
1 #include <stdio.h>
2
3 int main(){
4     int a, b , c; //representan tres numero enteros
5
6     scanf("%d %d %d",&a,&b, &c);
7
8     if( a > b ){
9         if(b > c){
10             printf("%d<%d<%d\n",c,b,a);
11         }else //c > b {
12             if(a > c){
13                 printf("%d<%d<%d\n",c,b,a);
14             }else{
15                 printf("%d<%d<%d\n",b,a,c);
16             }
17         }else{
18             if(a > c){
19                 printf("%d<%d<%d\n",c,a,b);
20             }else{
21                 if( b > c){
22                     printf("%d<%d<%d\n",a,c,b);
23                 }else{
24                     printf("%d<%d<%d\n",a,b,c);
25                 }
26             }
27         }
28
29     return 0;
30 }
```

```
1 #include <stdio.h>
2
3 int main(){
4     int a, b , c; //representan tres numero enteros
5
6     scanf("%d %d %d",&a,&b, &c);
7
8     if( a > b ){
9         if(a > c){
10             if(b < c){
11                 printf("%d<%d<%d\n",c,b,a);
12             }else{
13                 printf("%d<%d<%d\n",b,c,a);
14             }
15         }else{
16             if(b > c){
17                 printf("%d<%d<%d\n",a,c,b);
18             }else{
19                 printf("%d<%d<%d\n",a,b,c);
20             }
21         }
22     }else{
23         if(b > c){
24             if( a > c ){
25                 printf("%d<%d<%d\n",c,a,b);
26             }else{
27                 printf("%d<%d<%d",a,c,b);
28             }
29         }else{
30             if(a > c){
31                 printf("%d<%d<%d\n",b,c,a);
32             }else{
33                 printf("%d<%d<%d\n",b,a,c);
34             }
35         }
36     }
37 }
```

7. Y para finalizar en el ejercicio 10 usamos la estructura switch, que para indicar la finalización de cada caso hay que usar la instrucción break.

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(){
5      int x, mod;
6      float fx;
7      scanf("%d",&x);
8      mod = x % 4;
9
10     switch(mod){
11         case 0:
12             fx = x*x*x;
13             break;
14         case 1:
15             fx = sqrt(x);
16             break;
17         case 2:
18             fx = exp(x);
19             break;
20         case 3:
21             fx = 1/x;
22             break;
23     }
24     printf("%f\n", fx);
25     return 0;
26 }
```

Conclusiones

Esta práctica me pareció bastante sencilla y clara. Analizamos la sintaxis de cada estructura de selección y la razón de la simplicidad es que estas estructuras son muy similares en la sintaxis al pseudocódigo, entonces resultó más sencillo realizar códigos con estas. Me pareció bastante interesante la estructura ternaria, ya que cuando lleguemos a construir programas más complejos y con más esta será una buena herramienta para ahorrarse unas cuantas líneas; así como sucede con la combinación de operadores booleanos, con los que podemos evaluar una condición con el resultado de una función previa. -Walls Chávez Luis Fernando

Las estructuras de selección vistas en la práctica , fueron analizadas y tomadas de forma similar que en Pseudocódigo, por lo cual solo se reforzó de una forma mas clara y formal el uso de estas estructuras en lenguaje C, las mayores dificultades solo se mostraron en la forma de sintaxis o errores que no hay que dejar pasar como el famoso “;” o alguna llave , entre otros.

Gracias a lo visto en clase y lo anteriormente aprendido, fue mucho mas fácil poder realizar las actividades dadas en está práctica.

-Morales Velasco Brayan

En esta práctica utilizamos las estructuras de selección, que nos permiten resolver problemas con diferentes resultados en el lenguaje C. Es necesario saber que estructura elegir para cada caso, y realizar la sintaxis correcta, ya que un mínimo error o alguna omisión que hagamos, puede generar que el programa no compile.

Estas estructuras ya las conocíamos con los ejercicios en pseudocódigo que realizamos, pero es aún más importante saberlas utilizar en C.

- Lagunas Parra Jaime Rodrigo