

Programación I

Práctica #2

Curso 2023-2024

1º curso

Las instrucciones de decisión en C

Introducción

En la Práctica #1 hemos desarrollado programas sencillos en los que los pasos a realizar por dicho programa son siempre los mismos y no se repiten. Es decir, los algoritmos que hemos implementado son secuencias lineales formadas por un conjunto de operaciones fijo.

A medida que los algoritmos que queremos programar se van complicando, esto deja de ser así: tendremos operaciones que se realizarán sólo cuando se den determinadas condiciones (ejecución **condicional**) o bien tendremos instrucciones que queremos que se repitan varias veces (ejecución repetitiva).

En esta práctica aprenderemos las construcciones que el lenguaje C nos proporciona para codificar la ejecución condicional, es decir, las instrucciones de **decisión**, también llamadas de **selección**.

Objetivos



Esta práctica tiene como objetivos fundamentales:

- Profundizar en los tipos básicos de datos del lenguaje C.
- Aprender a utilizar las diferentes estructuras de control de flujo del lenguaje C: ejecución condicional (if-else y switch).
- Continuar adquiriendo destreza en el proceso de desarrollo de programas C.

Planificación

Sesión 2 de laboratorio (3 horas)

1. Realización de los ejercicios 1 a 3.
2. Realización de los ejercicios 4 y 5.
3. Realización de los ejercicios 6 y 7.

Trabajo fuera del aula:

- Terminar los ejercicios no completados en el aula.
- Seguir con el diseño y codificación de los programas indicados en los anexos I y II de Práctica #1.

Instrucciones de decisión

La instrucción if-else

La instrucción de selección if-else se utiliza típicamente cuando queremos ejecutar un bloque de código diferente en función de que se cumpla o no una determinada condición.

La instrucción if-else tiene la siguiente sintaxis:

```
if (<expresión>) {  
    <bloque de instrucciones 1>  
}  
else {  
    <bloque de instrucciones 2>  
}
```

El bloque de instrucciones 1 se ejecutará si la expresión se evalúa a un valor distinto de cero y el bloque 2 si la expresión se evalúa a cero.

Por ejemplo, si queremos mostrar el mayor de dos números enteros, podríamos hacer:

```
if (a > b) {  
    printf ("%d es mayor que %d\n", a, b);  
}  
else {  
    printf ("%d es mayor que %d\n", b, a);  
}
```

Una versión reducida de esta instrucción es aquella que no tiene cláusula else. En este caso tendremos un bloque de código que sólo se ejecutará si se cumple la condición indicada. La alternativa (no se cumple la condición) será no hacer nada.

Además, podemos encadenar varias instrucciones if-else para seleccionar un bloque de código a ejecutar de entre varios (normalmente un número pequeño de alternativas)

Ejercicio 1:

Escribe un programa en lenguaje C (ej1.c) que solicita al usuario que tome una decisión. Para ello, el programa muestra en pantalla el siguiente mensaje:

Are you sure you want to close the app? (YN):

Finalmente, el programa presentará uno de los siguientes mensajes:

- You have chosen "Yes"
- You have chosen "No"

El primero aparecerá si el usuario pulsa la tecla 'y' o la tecla 'Y', y el segundo si el usuario pulsa la tecla 'n' o la tecla 'N'. En ambos casos el programa terminará tras presentar el mensaje.

Consejo: usa las funciones toupper() o tolower().

Ejercicio 2:

Copia el programa del ejercicio 1:

```
usuario@maquina:~/pr02$ cp ej1.c ej2.c
```

Modifícalo para que presente un tercer mensaje:

- Invalid option

si la tecla pulsada no es ninguna de las reconocidas ('y', 'n', 'Y' o 'N')

Ejercicio 3:

Copia el ejercicio 7 de la Práctica #1:

```
usuario@maquina:~/pr02$ cp ../pr01/ej7.c ej3.c
```

Modifícalo para que, primero, la invitación a teclear indique que solo se admiten enteros con un valor entre 0 y 1000, y, después:

- si el entero introducido no está en ese rango, muestre el mensaje:

Valor incorrecto

- si está dentro, muestre la raíz de ese entero.

Por ejemplo:

Dame un entero [0-1000]: -7

Valor incorrecto

Pero:

Dame un entero [0-1000]: 75

RAIZ: 8.66

Ejercicio 4:

Copia el ejercicio 8 de la Práctica #1:

```
usuario@maquina:~/pr02$ cp ../pr01/ej8.c ej4.c
```

Modifícalo para que, primero, la invitación a teclear indique que solo se admiten los caracteres que forman parte de la cadena "EQLFX", y después, convierta el carácter leído a mayúsculas, y:

- si el carácter introducido forma parte de dicha cadena, lo muestre por pantalla,
- y, si no, muestre el mensaje:

No es valido

Por ejemplo:

Dame un caracter (EQLFX): q
La letra q en mayusculas es Q

Pero:

Dame un caracter (EQLFX): 2
No es valido

Para resolver este ejercicio necesitamos utilizar la función `strchr()`

Ejercicio 5:

Copia el ejercicio 9 de la Práctica #1:

```
usuario@maquina:~/pr02$ cp ../pr01/ej9.c ej5.c
```

Modifícalo para que, primero, la invitación a teclear indique que solo se admiten cadenas con hasta 24 caracteres, y, después:

- si la longitud de la cadena leída es mayor que 24 caracteres, muestre el mensaje:

Longitud excesiva

- si no, muestre la longitud de la cadena.

Por ejemplo:

Dame una cadena (1-24 char): Programacion_en_lenguaje_C
Longitud excesiva

Pero:

Dame una cadena (1-24 char): Programacion
La cadena tiene 12 caracteres

La instrucción switch

La instrucción switch también permite ejecutar bloques de código de forma condicional.

A diferencia de la instrucción if-else, la instrucción switch se suele utilizar cuando el número de alternativas a considerar es elevado.

En la instrucción switch, la selección del bloque de instrucciones que se va a ejecutar se realiza en función de los diferentes valores que puede tomar una expresión, que se debe evaluar a un valor de tipo int o a un valor tipo char.

La sintaxis más típica de la instrucción switch es la siguiente:

```
switch (<expresión>) {  
    case <valorA>:  
        <bloque de instrucciones 1>  
        break;  
    case <valorB>:  
    case <valorC>:  
        <bloque de instrucciones 2>  
        break;  
    ...  
    default:  
        <bloque de instrucciones N>  
        break;  
}
```

Por ejemplo:

```
switch (a) {  
    case 0:  
        printf ("a vale cero\n");  
        break;  
    case 1:  
        printf ("a vale uno\n");  
        break;  
    case 2:  
    case 3:  
    case 4:  
        printf ("a vale entre dos y cuatro\n");  
        break;  
    default:  
        printf ("a no nos vale\n");  
        break;  
}
```

Ejercicio 6:

Escribe un programa en lenguaje C (ej6.c), que muestre en pantalla lo siguiente:

E) Enrol
Q) Quit
L) List
F) Filter

Choose an option:

A continuación el programa leerá de teclado un valor de tipo char y mostrará **uno** de los siguientes mensajes según corresponda (con independencia del caso del carácter leído, es decir, de si está en mayúsculas o minúsculas):

Enrol
Quit
List
Filter
Invalid option

Ejercicio 7:

Escribe un programa calculadora (ej7.c). El programa debe invitar al usuario a introducir una operación, y después, leer por teclado, y **en una sola línea**:

- un número tipo float,
- uno de los siguientes caracteres: '+', '-', '*', '/',
- y otro número tipo float.

Finalmente, debe presentar en pantalla el resultado de la operación correspondiente sobre los dos números, con dos decimales.

Ejemplo:

Dame una operacion: 5.2 + 7.3
Resultado: 12.50

Resumen

Los principales resultados esperados de esta práctica son:

- Mejorar el dominio de los tipos básicos de datos del lenguaje C.
- Conocer el manejo de las instrucciones de decisión del lenguaje C.

Como posibles líneas de trabajo adicional (opcional) del alumno, se proponen las siguientes:

- Terminar los ejercicios no completados en clase.
- Seguir practicando con los ejercicios de los Anexos I y II de la Práctica #1