



Fundamentos de Programación
Grado en Ing. Informática

Guion práctico nº 2



Tema 3.- Estructuras de control

Sentencias Condicionales o de Selección

1. Diseñe un programa que lea una nota con decimales por teclado y muestre por pantalla la calificación que tiene.

NOTA	Mensaje
< 0	Error en la nota
[0, 5)	SUSPENSO
[5 y 7)	Aprobado
[7 y 9)	Notable
[9 y 10]	Sobresaliente
> 10	Error en la nota

2. Diseñe un programa que lea dos números enteros por teclado y determine cuál es el mayor y cuál es el menor. También deberá considerar el caso en el que los dos números sean iguales.
3. Diseñe un programa que resuelva una ecuación de segundo grado $ax^2+bx+c = 0$. Debe pedir al usuario que introduzca los valores de a, b y c de tipo entero y el programa indicará los valores de las raíces solución (una o dos raíces según corresponda), o bien un mensaje indicando que no tiene raíces reales.
4. Diseñe un programa que pida dos números enteros por teclado y muestre por pantalla el siguiente menú:

MENU

- 1. Sumar
- 2. Restar
- 3. Multiplicar
- 4. Dividir

Elija opción:

El usuario deberá elegir una opción y el programa mostrará el resultado por pantalla, o el mensaje **“Opción errónea”** si la opción seleccionada es incorrecta.

5. Implemente un programa que pida por teclado un año y muestre un mensaje por pantalla indicando si es bisiesto o no. **Nota:** Un año es bisiesto si es múltiplo de 400 o bien, si es múltiplo de 4 y no de 100.
6. Diseñar un programa que solicite por teclado dos números enteros y la operación a realizar con ellos:
- Si el usuario pulsa '+', 's' o 'S' el programa mostrará el resultado de sumar los números indicados.
 - Si el usuario pulsa '-', 'r' o 'R' mostrará la resta.
 - Para cualquier otra pulsación se mostrará el siguiente mensaje de error *“operación no disponible”*.

7. Implementar un programa que ayude a calcular los descuentos que aplica una agencia de viajes en sus billetes de avión. El programa comenzará pidiendo los siguientes datos:

- Distancia del viaje a realizar.
- Número de días que dura dicho viaje.
- Edad del turista.

A continuación mostrará el precio del viaje sin descuento y un mensaje indicando si tiene derecho a un descuento que, en caso afirmativo, irá acompañado de la cantidad a la que asciende dicho descuento y del precio final del viaje.

El precio del viaje se obtiene en función de los kilómetros recorridos, a razón de 0.75€/km. Una persona tendrá derecho a una reducción del 25% del precio del billete si el nº de días de estancia es superior a 7 y la distancia superior a 800km, o bien si es mayor de 55 años.

8. Modificar el código solución al ejercicio 24 del guion práctico nº 1, de modo que el usuario pueda introducir el valor de la carga en cualquiera de las siguientes unidades:

- a) *Miliculombios* ($mC = 10^{-3}C$)
- b) *Microculombios* ($\mu C = 10^{-6}C$)
- c) *Nanoculombios* ($nC = 10^{-9}C$)

La distancia también podrá ser suministrada por parte del usuario en:

- a) *Decímetros* (dm)
- b) *Centímetros* (cm)
- c) *Milímetros* (mm)

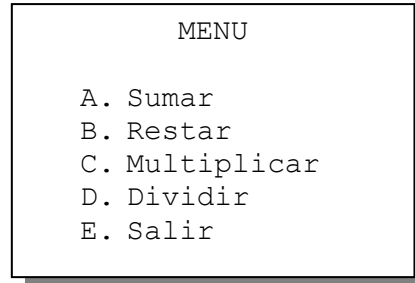
Cuando el usuario indique el valor numérico de la carga, el programa le ofrecerá mediante un menú las diferentes opciones en cuanto a unidades, para que seleccione la correcta. Del mismo modo se trabajará con la distancia. Si el usuario pulsa una opción incorrecta se tomará por defecto la unidad de almacenamiento (μC y m) y se le avisará con un mensaje.

La siguiente tabla puede ser utilizada por el alumno para comprobar los resultados obtenidos:

$q = 1 \times 10^{-6} mC$	$r = 20 \text{ mm}$	$E = 2.25 \times 10^4 \text{ N/C}$
$q = 5 \text{ nC}$	$r = 30 \text{ cm}$	$E = 499.5 \text{ N/C}$
$q = 0.04 \mu C$	$r = 20 \text{ dm}$	$E = 89.9 \text{ N/C}$

Sentencias Iterativas o Bucles

1. Modifique la solución aportada al ejercicio 4 relacionado con las sentencias de selección, de tal forma que las opciones se escojan por letra y se añada una nueva (E) que servirá para salir del programa:



El programa una vez realizada una opción deberá mostrar el menú de nuevo hasta que se elija la opción E (Salir).

2. Realizar un programa que pida por teclado cuántos números enteros se van a introducir y a continuación los solicite por teclado, para calcular el valor máximo, el mínimo y la media de todos ellos.
3. Implementar un programa que pida por teclado cuántos alumnos se han examinado y sus notas, que deben estar comprendidas en el rango [0,10] (si esto no se cumple se volverá a solicitar esa nota hasta que sea correcta). El programa mostrará por pantalla una estadística que indique el nº de suspensos, aprobados, notables y sobresalientes que ha habido en el examen.
4. Diseñar un programa que muestre por pantalla en formato 2D la tabla de multiplicar del número indicado por el usuario. Si el valor introducido es menor que 1 o mayor que 10 el programa mostrará un mensaje y volverá a solicitar otro número.
5. Implementar un programa para mostrar varias tablas a la vez de manera horizontal. En primer lugar se debe solicitar por teclado el número de la tabla inicial y de la final. Los valores indicados deben estar en el rango [1, 10] de lo contrario se solicitarán de nuevo. En el caso de que el número de la tabla inicial sea mayor que el de la tabla final, deberán ser intercambiados.

Ejemplo de ejecución:

Indique tabla inicial y final: 8 5			
5 x 0 = 0	6 x 0 = 0	7 x 0 = 0	8 x 0 = 0
5 x 1 = 5	6 x 1 = 6	7 x 1 = 7	8 x 1 = 8
.
.
5 x 10 = 50	6 x 10 = 60	7 x 10 = 70	8 x 10 = 80

Nota: Utilizar en la sentencia `cout` la constante `\n` para ir a una nueva línea y `\t` para introducir un tabulador.

6. Implementar un programa que pida por teclado un número y muestre por pantalla todos los divisores de dicho número.
7. Implementar un programa que pida un número N por teclado y determine si es o no un número abundante y en caso afirmativo mostrar su abundancia. Definiciones:
 - Sea S la suma de todos los divisores de N . Se dice que N es un número abundante si cumple que S es mayor estricto que el doble de N ($S > 2 \times N$).
 - Se define la abundancia de N como el valor $S - 2 \times N$.
8. Implementar un programa que pida por teclado un número entero e indique si es o no un número primo
9. Modificar la solución al ejercicio anterior, para que muestre por pantalla los números primos comprendidos en un intervalo solicitado al usuario.
10. Realizar un programa que pida por teclado un número entero que debe ser positivo y menor que 20 (en caso contrario se volverá a solicitar hasta que se cumpla la condición) y devuelva el factorial de dicho número.
11. Implementar un programa que pregunte al usuario cuántos números va a introducir y devuelva la suma de todos los no primos. Se ha de controlar que la cantidad de números indicada sea mayor que cero, ya que de lo contrario no se aceptará y habrá que volverla a solicitar.
12. Diseñe un programa que pida dos valores enteros, a y b , por teclado y devuelva su máximo común divisor (**mcd**), aplicando el algoritmo de Euclides.

Dicho algoritmo dice que si a es mayor que b entonces $a = a - b$ y si b es mayor que a entonces $b = b - a$ y esto se repite hasta que a y b son iguales, momento en el que el máximo común divisor es a

Por ejemplo

$a = 6, b = 8$	Como $a < b$, entonces $b = b - a, b = 8 - 6 = 2$
$a = 6, b = 2$	Como $a > b$, entonces $a = a - b, a = 6 - 2 = 4$
$a = 4, b = 2$	Como $a > b$, entonces $a = a - b, a = 4 - 2 = 2$
$a = 2, b = 2$	Como $a = b$, entonces el mcd es 2

13. Diseñar un programa que muestre por pantalla tantos términos de la sucesión de Fibonacci como haya solicitado el usuario. Una cantidad de términos menor que 1 no será aceptada, de manera que se le indicará con un mensaje al usuario y se le volverá a pedir otra. El programa, además de mostrar la secuencia de Fibonacci, deberá indicar la media de todos sus elementos.

Nota: La sucesión de Fibonacci es aquella cuyos dos primeros elementos son el 0 y el 1 y los siguientes se obtienen como suma de los dos elementos inmediatamente anteriores. Por ejemplo:

Orden	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Valor	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610

14. Modificar la solución al ejercicio anterior, añadiéndole una funcionalidad adicional que consistirá en solicitar un número por teclado y comprobar si es un elemento de la secuencia de Fibonacci generada. En caso afirmativo devolverá la posición que ocupa dentro de la secuencia y en caso contrario se mostrará el mensaje *“El nº introducido no pertenece a la sucesión”*. Por ejemplo, si introducimos el nº 34, nos devolverá la posición 10, pero si introducimos el valor 40 nos mostrará el mensaje.
15. Modificar el código solución para el ejercicio nº 23 del guion práctico nº 1, de modo que indique si la *Fuerza* es de **atracción** (valor negativo para F) o de **repulsión** (valor positivo para F).

Tras mostrar un resultado al usuario, el programa preguntará si se quiere volver a ejecutar con otros datos diferentes, de manera que:

- Si la respuesta es afirmativa se le pedirán nuevos valores para las cargas y la distancia entre ellas.
- Si la respuesta es negativa se le mostrará un mensaje de agradecimiento por el uso del software.

La siguiente tabla puede ser utilizada por el alumno para comprobar los resultados obtenidos:

$q_1 = 3\mu\text{C}$	$q_2 = -8\mu\text{C}$	$r = 2\text{m}$	$F = -0.054\text{ N}$ (Atracción)
$q_1 = -0.5\mu\text{C}$	$q_2 = 644\mu\text{C}$	$r = 3.5\text{m}$	$F = -0.237\text{ N}$ (Atracción)
$q_1 = 2.8\mu\text{C}$	$q_2 = 7.5\mu\text{C}$	$r = 0.1374\text{m}$	$F = 10\text{ N}$ (Repulsión)
$q_1 = 4\mu\text{C}$	$q_2 = -8\mu\text{C}$	$r = 0.004\text{m}$	$F = -18000\text{ N}$ (Atracción)
$q_1 = 1\mu\text{C}$	$q_2 = 2.5\mu\text{C}$	$r = 0.05\text{m}$	$F = 9\text{ N}$ (Repulsión)

16. Se pretende simular el funcionamiento de un **reloj digital** de manera que comience su funcionamiento a la hora indicada por el usuario (p. ej. las **12:58:0**) y realice la cantidad de **minutos de simulación** que se le haya solicitado desde teclado (p.ej. 3 minutos de simulación implicarían acabar a las **13:0:59**).

Para ello, se le pedirá al usuario la hora y el minuto de inicio de la simulación y se pondrá a 0 la cantidad de segundos de inicio. A continuación se solicitará la cantidad de tiempo, en minutos, que deberá durar dicha simulación, por lo que tendrá que ser un valor **mayor o igual que cero** (de no ser así se le indicará esta circunstancia al usuario con un mensaje y se le volverá a solicitar otra cantidad de minutos).

Una vez solicitados los datos necesarios, se mostrará el reloj por pantalla cambiando tal y como se ha indicado y realizará el tiempo de simulación solicitado. Comenzará con la hora de inicio, **Hi:Mi:Si**, y por cada segundo de simulación se volverá a mostrar de nuevo, borrando previamente el anterior.

Para simular un tiempo de espera de un segundo, utilizaremos la función **Sleep(milisegundos)** que se encuentra en la librería **windows.h**. Se aconseja al alumno que para que la simulación no sea ni muy rápida ni excesivamente lenta, utilice la función Sleep con 500 milisegundos, **Sleep(500)**; de manera que para la simulación 1 segundo serán realmente 500 milisegundos.

17. Implementar un programa que simule un juego de acertar un número de tres cifras. El programa generará un número al azar y el usuario deberá intentar acertarlo con un máximo de 10 intentos. En cada intento el programa indicará si el número introducido por el usuario es mayor, menor o ha acertado.

```
//Inicializar los números aleatorios  
srand(time(NULL));  
// Número aleatorio entre 100 y 999  
numero=100+rand()%(999-100);
```

18. Se quiere codificar un juego consistente en adivinar un número. Para ello se pide implementar un programa que realice las siguientes acciones:

- a) **Establecer parámetros del juego:** Se determinará cuál será el número secreto y con qué cantidad de puntos comienza el jugador la partida.

Para obtener el número secreto, se le solicitará al jugador que indique un número. Dicho número será la **semilla** a partir de la cual se calculará el número secreto del siguiente modo:

- 1) Si el valor obtenido de la siguiente operación **(semilla%90) x123** es un **número abundante** (ver ejercicio 7), entonces el **número secreto será** el que se obtenga tras la ejecución de la siguiente expresión **((semilla%90) x123) %12459**.
- 2) En caso contrario (que **no** es **abundante**) el número secreto será **((semilla%90) x123) %5397**.

- b) **Jugar:** Se implementará el juego, consistente en pedir al jugador que intente adivinar el número secreto, teniendo en cuenta:

- 1) Si el número indicado no es el secreto y se ha quedado corto se le quita 1 punto, pero si se pasa se le quitan 2.
- 2) El juego finalizará cuando se adivina el número o cuando el jugador se queda sin puntos. En cualquier caso se le mostrará mensaje que indique lo ocurrido, que será de enhorabuena con los puntos conseguidos, o lamentando que haya perdido.

19. Juego "**Te adivino el número que estás pensando**". En este caso se trata de programar un juego para que sea el ordenador el que te haga preguntas para intentar adivinar el número que has pensado previamente. Para ello:

- a) Se solicitará por teclado los extremos **[a,b]** de un rango en el que el usuario debe pensar un número. El extremo **a** debe ser menor que **b**, de lo contrario se mostrará por pantalla un error y se volverán a pedir.
- b) Para intentar adivinar el número, el programa calculará la media aritmética de los valores extremos del intervalo, la mostrará por pantalla y preguntará por teclado si el valor pensado por el jugador es menor (<), mayor (>), o igual (=) al propuesto:
 - Si el usuario contesta con un símbolo diferente a los tres posibles, mostrará un mensaje de error y volverá a realizar la misma pregunta.
 - Si el usuario dice que el número que ha pensado es menor (<) al indicado por el software, **b** (el límite superior del rango) pasará a ser **media-1**.
 - Si el usuario dice que el número que ha pensado es mayor (>) al indicado por el software, **a** (el límite inferior del rango) pasará a ser **media+1**.
 - Cuando el usuario indica que ha acertado (=) el programa mostrará "He ganado".

- c) Si se detecta que el usuario intenta engañar, también habrá de mostrarse un mensaje con lo ocurrido. El usuario estará engañando cuando al calcular los nuevos límites del rango, se comprueba que el inferior es mayor que el superior.

El juego finaliza cuando el programa adivina el número o detecta que el usuario le está engañando.

20. Realizar un programa que calcule el módulo de Young, que se define como $E = s/e$, por lo que será necesario conocer el valor del esfuerzo actuante (s) y la deformación del material (e). Estos se calculan como:

Esfuerzo actuante:

$$s = \frac{F}{A_0} = + \frac{F(Newtons)}{\Pi \left(\frac{Sección(mm)}{2} \right)^2}$$

Deformación del material:

$$e = \frac{\Delta l}{l_0}$$

El programa comenzará solicitando los datos de: carga, sección, longitud inicial y el incremento de la longitud.

Cada vez que se introduzca un dato el programa preguntará si el valor ha sido indicado en **Newtons** o **KiloNewtons** para la carga, y en metros o **milímetros** para las longitudes y sección. Hay que tener en cuenta que los cálculos se deben hacer en **Newtons** y en **milímetros**, por lo que es posible tener que hacer una conversión de unidades. El resultado del módulo de Young hay que darlo en **MegaPascles**.

Veámoslo con un ejemplo:

Suponga un objeto metálico de sección circular de 10 mm de diámetro y 60 mm de longitud que, al aplicarle un carga de 50 kN, se alarga elásticamente 0.62 mm. Calcule el módulo de Young, que se utiliza para saber el tipo de material que se ha utilizado en la construcción.

El módulo de Young se define como $E = s/e$, por lo que será necesario conocer el valor del esfuerzo actuante (s) y el índice de deformación del material (e). Estos se calculan como:

$$s = \frac{F}{A_0} = \frac{50000N}{\Pi \left(\frac{10mm}{2} \right)^2} = 636.62MPa \quad e = \frac{\Delta l}{l_0} = \frac{(0.62mm)}{(60mm)} = 0.0103$$

Por tanto:

$$E = \frac{(636.62MPa)}{(0.0103)} = 61807.77MPa = 61.8GPa$$