



UNIVERSIDAD DE MÁLAGA
DPTO. DE LENGUAJES Y C. DE LA COMPUTACION
E.T.S. DE INGENIERIA INFORMATICA

FUNDAMENTOS DE LA PROGRAMACIÓN

PRIMERA RELACIÓN DE PROBLEMAS

1. Dado **int** num, escribe una expresión booleana que determine si num es un número par.
2. Dado **int** num, escribe una expresión booleana que determine si num es un número de tres dígitos.
3. Dado **int** num, escribe una expresión booleana que determine si num es un divisor de 100.
4. Escribe una expresión booleana que represente la pertenencia: $x \in \{3, 4, 5, 6, 7\}$.
5. Escribe una expresión booleana que represente la pertenencia: $x \in \{1, 2, 3, 7, 8, 9\}$.
6. Escribe una expresión booleana que represente las pertenencias: $x \in \{3, 4, 6, 8, 9\}$, $y \in \{6, 7, 8, 3\}$
7. Dados **int** x, y, escribe una expresión booleana que represente que ni x ni y son mayores que 10.
8. Dado **char** c, escribe una expresión booleana que determine si c es una letra mayúscula.
9. Dado **char** c, escribe una expresión booleana que determine si c es una letra.
10. Mejora el siguiente fragmento de código, siendo b una expresión lógica y T, S y U tres sentencias cualesquiera que no afectan a b:

```
if (b) {  
    T;  
    S;  
} else {  
    T;  
    U;  
}
```

11. Mejora el siguiente fragmento de código, siendo b una expresión lógica y x una variable booleana:

```
if (b) {  
    x = true;  
} else {  
    x = false;  
}
```

12. Simplifica el siguiente segmento de algoritmo de forma que se necesiten menos comparaciones:

```

if (edad >= 65) {
    cout << "Jubilado";
} else {
    if (edad < 18) {
        cout << "Menor de edad";
    } else {
        if ((edad >= 18) && (edad < 65)) {
            cout << "Activo";
        }
    }
}

```

13. Diseña un algoritmo que lea por teclado dos números enteros y determine si el primero es divisor del segundo, si el segundo es divisor del primero o si no ocurre nada de eso. Por ejemplo, si se introducen los números 8 y 2, la salida indicará que 2 es divisor de 8. En cambio, si se introducen los números 5 y 7, la salida indicará que los números no son divisibles.
14. Diseña un algoritmo para determinar si un número entero (leído por teclado) es múltiplo de alguno de los siguientes números (o de varios de ellos): 3, 4, 5. No es necesario indicar de qué números en concreto es múltiplo. Por ejemplo, si se introduce el número 12, la salida indicará que sí es divisible. En cambio, si se introduce el número 17, la salida indicará que no es divisible.
15. Escribe un algoritmo para calcular el precio de una partida de artículos aplicando un descuento creciente con la cantidad comprada según la siguiente tabla (la cantidad comprada se lee por teclado):

Número de unidades	Precio unitario (euros)
1	100
2	95
3	90
4 o más	85

Así, por ejemplo, si se introduce una cantidad de 3 unidades, la salida será de 270 euros (resultado de multiplicar 3 por 90). Si se introduce una cantidad de 7 unidades, la salida será de 595 euros (resultado de multiplicar 7 por 85).

16. ¿Cuántas veces se ejecutará la Acción I ?

```

cont=0;
while (cont < 10) {
    cont = cont + 1;
    if (cont == 10) {
        Acción I
    }
}

```

17. ¿Cuál es el valor de la variable booleana test después de la ejecución de los siguientes bucles?

a.

```
bool test;
int N;

test = true;
for (int contador= 1;contador<=N;contador++) {
    test = !test;
}
```

b.

```
bool test;
int N;

test = N %2 ==0;
for (int contador= 1;contador<=N;contador++) {
    test = !test;
}
```

18. ¿Qué rango de valores enteros pueden tomar X e Y para garantizar que el siguiente algoritmo iterativo termina?

```
A = X;
B = Y;
do {
    while (A > B) {
        A = A - B;
    }
    while (B > A) {
        B = B - A;
    }
} while (A != B);
```

19. Confecciona un algoritmo que lea de teclado un texto carácter a carácter hasta localizar un punto, y que al final dé como salida el número de comas encontradas, y el número de caracteres leídos. Por ejemplo, si se introduce el texto siguiente:

Este, aunque sencillo, es un ejemplo interesante.

La salida será:

Caracteres: 48

De ellos son comas: 2

20. Diseña un programa que lea un único número natural por teclado (hay que controlar que sea mayor o igual que 0) en una variable de tipo int y muestre por pantalla cuantos dígitos pares tiene. Por ejemplo: 3445621 tiene 4 dígitos pares; 55914 tiene 1 dígito par. NOTA: El dígito más a la derecha de un número puede obtenerse calculando el resto de la división por 10 y el número sin el dígito más a la derecha puede obtenerse dividiendo el número por 10. Ejemplo: $123 \% 10 = 3$; $123 / 10 = 12$.

21. Diseña un algoritmo que lea un número entero n por teclado (hay que controlar que sea distinto de 0). Después se le pedirá al usuario que introducirá por teclado una secuencia de números enteros terminada en 0. El algoritmo debe determinar si el número n aparece o no en la secuencia. Para ello, el algoritmo leerá números de la secuencia hasta encontrar el número n o bien hasta leer el 0, mostrando por pantalla el mensaje correspondiente. En el caso de que se detecte el número buscado en la secuencia ya no se seguirán leyendo más números. Dos ejemplos de ejecución serían:

Introduzca el numero entero a buscar (distinto de 0): 3

Introduzca una secuencia de numeros enteros terminada en 0: 4 2 3 -5 7 -6 0

El numero 3 SI aparece en la secuencia

Introduzca el numero entero a buscar (distinto de 0): 3

Introduzca una secuencia de numeros enteros terminada en 0: 4 2 -3 -5 7 -6 0

El numero 3 NO aparece en la secuencia

22. Diseña un algoritmo que determine si la cadena abc aparece en una sucesión de caracteres introducida por el usuario y cuyo final viene dado por un punto. El algoritmo leerá caracteres hasta encontrar la cadena abc o bien hasta leer el punto, mostrando por pantalla el mensaje correspondiente. Al igual que en el problema anterior, en el caso de que se detecte la presencia de la cadena buscada, ya no se seguirán leyendo más caracteres. Por ejemplo, si se introduce el texto hpabxbcpq., la salida será Cadena abc NO encontrada. En cambio, si se introduce el texto haabcbxbcpq., la salida será Cadena abc SI encontrada.
23. Diseña un algoritmo que lea un número natural n por teclado (hay que controlar que sea mayor que 0) y calcule el n-ésimo número de la serie de Fibonacci. Los dos primeros números de esta serie son el cero y el uno, y a partir de éstos cada número se calcula realizando la suma de los dos anteriores. Por ejemplo, si se introduce el número 5, la salida será 3, que es el 5º número de la serie. Si se introduce el número 9, la salida será 21, que es el 9º número de la serie.
24. Escribe un algoritmo que encuentre el mayor, el menor y la media aritmética de una colección de N números enteros leídos por el teclado donde N es el primero de los números. Un ejemplo de ejecución sería:

Introduzca la secuencia de numeros: 5 3 7 2 1

El mayor es: 7

El menor es: 1

La media es: 3.6

25. Escribe un algoritmo que lea una lista de números enteros terminada en 0 y que encuentre y escriba en la pantalla la posición de la primera y de la última aparición del número 12 dentro de la lista. Ejemplos de ejecución:

Introduzca la secuencia de numeros (0 para terminar): 3 -2 4 6 0

Primera Aparicion del 12: 0

Ultima Aparicion del 12: 0

Introduzca la secuencia de numeros (0 para terminar): 3 12 4 6 0

Primera Aparicion del 12: 2

Ultima Aparicion del 12: 2

Introduzca la secuencia de numeros (0 para terminar): 3 12 4 12 -5 8 12 0

Primera Aparicion del 12: 2

Ultima Aparicion del 12: 7

26. Un número se dice que es perfecto si su valor coincide con la suma de sus divisores sin contar al propio número. Así, 6 es perfecto porque $6 = 1 + 2 + 3$ y 28 también lo es porque $28 = 1 + 2 + 4 + 7 + 14$. Diseña un algoritmo que muestre por pantalla el siguiente número perfecto después del 28.

27. Desarrolla un algoritmo para el siguiente juego entre un usuario y la máquina:

Se leen dos números enteros que el usuario introduce por teclado y que representan el límite inferior y superior de un determinado rango de valores. Después el usuario piensa un número en ese rango. La máquina tiene que acertar dicho número. Para ello propone un número y el usuario responde con un carácter: $>$ (si el número pensado es mayor que el propuesto), $<$ (si el número pensado es menor que el propuesto) o $=$ (si el número pensado es igual al propuesto). Si la respuesta es $>$ o $<$, la máquina propondrá otro número. El proceso se repite hasta que la máquina acierte el número pensado por el usuario.

28. Desarrolla un algoritmo para el siguiente juego entre un usuario y la máquina:

El usuario introduce un número natural n que representa a n objetos. La máquina decide quién empieza el juego y alternativamente (el usuario y la máquina) retiran cada uno 1, 2 o 3 objetos (es decir, elige restar 1, 2 o 3 a n). El que retira el último objeto pierde.

En las condiciones del problema, es posible desarrollar un algoritmo para que siempre gane la máquina. Construye tal algoritmo.