

Ejercicios sobre recursividad

1. Escribe una función recursiva que calcule el factorial de un número entero pasado como parámetro.
2. Completa el código de la siguiente función para que, de forma recursiva, calcule y devuelva el valor de a elevado a b.

```
int potencia( int a, int b )
```

3. Completa el código de la siguiente función para que, de forma recursiva, calcule y devuelva el número de Fibonacci de n.

```
int fibonacci( int n )
```

4. Escribe una función recursiva que, dado un número entero mayor que cero, muestre todos los enteros comprendidos entre 0 y él (ambos incluidos) en orden creciente.
5. Escribe una función recursiva que, dado un número entero mayor que cero, muestre todos los enteros comprendidos entre él y 0 (ambos incluidos) en orden decreciente.
6. Escribe una función recursiva que calcule el valor de $f(x)$ para cualquier $x \geq 0$ siendo $f(x) = f(x-2)*2 + f(x-3)*3$ si x es mayor que 3 y $f(x) = 2*x$ si x es menor o igual que 3 y mayor o igual que cero.
7. Escribe una función recursiva que, dado un número entero mayor que cero, muestre sus dígitos empezando por el de más a la derecha.
8. Escribe una función recursiva que, dado un número entero mayor que cero, calcule la suma de todos los enteros positivos menores o igual que él.
9. Escribe una función recursiva que sume los dígitos de un número entero mayor que cero pasado como parámetro
10. Escribe una función recursiva que, dado un número entero mayor que cero, muestre sus dígitos empezando por el de más a la izquierda.

Ejercicios de recursividad con cadenas de caracteres

11. Completa el código de la siguiente función para que, de forma recursiva, calcule la longitud de la cadena pasada como parámetro

```
int longitud( char * cadena )
```

12. Completa el código de la siguiente función para que, de forma recursiva, devuelva 1 si las dos cadenas pasadas como parámetro son iguales y 0 si son distintas.

```
int compara( char * a, char * b )
```

13. Completa el código de la siguiente función para que, de forma recursiva, devuelva 1 si la cadena pasada como parámetro es un palíndromo y 0 en caso contrario. Un palíndromo es una frase que se lee igual de izquierda a derecha que de derecha a izquierda, por ejemplo, la palabra "reconocer".

```
int palindromo( char * cadena )
```

14. Completa el código de la siguiente función para que, de forma recursiva, sustituya en la cadena pasada como parámetro todas las apariciones de la letra a por la letra b.

```
void sustituye( char * cadena, char a, char b )
```

15. Escribe una función recursiva que, dada una cadena, muestre por pantalla una secuencia de cadenas de longitud creciente empezando por la última letra y progresando hasta ser igual a la cadena original. Por ejemplo, para la cadena "Hola" se debe mostrar:

```
a
la
ola
Hola
```

16. Escribe una función recursiva que, dada una cadena, muestre por pantalla una secuencia de cadenas de longitud decreciente en la que cada una tenga una letra menos que la anterior. Por ejemplo, para la cadena "Hola" se debe mostrar:

```
Hola
ola
la
a
```

Ejercicios de recursividad con estructuras enlazadas

Usando las siguientes definiciones para implementar una estructura enlazada lineal

```
struct Nodo {  
    Elemento elem;  
    struct Nodo * sig;  
};  
  
typedef struct Nodo * NodoPtr;
```

17. Completa el código de la siguiente función para que, de forma recursiva, calcule la longitud de la estructura enlazada lineal sin cabecera pasada como parámetro.

```
int longitud( NodoPtr lista )
```

18. Completa el código de la siguiente función para que, de forma recursiva, muestre todos los elementos de la lista en el orden en el que están.

```
void muestra( NodoPtr lista )
```

19. Completa el código de la siguiente función para que, de forma recursiva, muestre todos los elementos de la lista en el orden inverso al que están.

```
void artseum( NodoPtr lista )
```

20. Completa el código de la siguiente función para que, de forma recursiva, libere toda la memoria asociada a la lista recibida como parámetro.

```
void libera( NodoPtr lista )
```

21. Completa el código de la siguiente función para que, de forma recursiva, calcule y devuelva 1 si las dos listas son iguales y 0 en caso contrario.

```
int compara( NodoPtr a, NodoPtr b )
```