

## Ejercicios de cuarto parcialito

### Del recuperatorio del quinto parcialito del segundo cuatrimestre de 2008

1) Escribir una función `merge_sort_3` que implemente un algoritmo similar al de merge sort pero que en lugar de dividir los valores en dos partes iguales, los divida en tres. Considerar que se cuenta con la función `merge(lista_1, lista_2, lista_3)`. ¿Cómo te parece que se va a comportar este método con respecto al merge sort original?

### Del primer recuperatorio del quinto parcialito del primer cuatrimestre de 2009

2) Escribir una función recursiva que reciba una lista y un parámetro `n`, y devuelva otra lista con los elementos de la lista replicados esa cantidad `n` de veces.

Por ejemplo, replicar `([1, 3, 3, 7], 2)` debe devolver `([1, 1, 3, 3, 3, 3, 7, 7])`.

### Del quinto parcialito del primer cuatrimestre de 2010

3) a) Mostrar paso a paso como funcionarían los algoritmos de inserción y quicksort sobre la siguiente lista de números: `[9, 12, 1, 4, 2, 8, 15, 21]`

b) ¿Cual de los dos algoritmos es más eficiente para ordenar una gran cantidad de elementos? ¿Por qué?

### Del cuarto parcialito del primer cuatrimestre de 2012

4) a) ¿Cómo debería ser un arreglo con números del 1 al 10 para obtener la peor performance en una implementación de quicksort que elige siempre como pivote al último elemento de la lista en vez del primero? Justifique.

b) ¿Qué método de ordenamiento elegiría para ordenar ascendentemente un arreglo que ya está ordenado pero en forma descendente? ¿Por qué?. Asumiendo que fueran muchos elementos, ¿elegiría este método de ordenamiento o utilizaría una función para invertirlos in-place (en el mismo arreglo)?

### Del quinto parcialito del primer cuatrimestre 2012

5) Escribir en C una función que pida al usuario que ingrese una cadena y la imprima invertida.

### Del segundo recuperatorio del quinto parcialito del primer cuatrimestre 2012

6) Escribir en C un programa que pida al usuario un valor mínimo, un valor máximo y un número `n`, e imprima una tabla con los cuadrados de los números entre mínimo y máximo cada `n` números. Por ejemplo (mínimo = 0, máximo = 17, `n` = 5) debe mostrar:

```
0 0
5 25
10 100
15 225
```

## Del cuarto parcialito del primer cuatrimestre de 2013

7) Escribir en C un programa que le solicite un número entero al usuario y muestre por pantalla si el número ingresado es un número primo o no.

## Del primer recuperatorio del tercer parcialito del primer cuatrimestre de 2013

8) Escribir una función recursiva que reciba una cadena y devuelva True si es un palíndromo (se lee igual hacia adelante que hacia atrás) o False si no lo es.

## Del segundo recuperatorio del cuarto parcialito del primer cuatrimestre de 2013

9) Escribir en C un programa que pida al usuario dos palabras. El programa debe imprimir ambas palabras en una línea, separadas por una secuencia de puntos de forma tal que la longitud total de la línea sea de 30 caracteres. Ejemplo:

Primera palabra: Hola  
Segunda palabra: Mundo

Hola.....Mundo

10) Escribir en C un programa que pida al usuario que ingrese 10 números enteros e imprima el mínimo, el máximo y el promedio.

## Del segundo recuperatorio del cuarto parcialito del primer cuatrimestre de 2015

11) El triángulo de Pascal es un arreglo triangular de números que se define de la siguiente manera: Las filas se enumeran desde  $n = 0$ , de arriba hacia abajo. Los valores de cada fila se enumeran desde  $k = 0$  (de izquierda a derecha). Los valores que se encuentran en los bordes del triángulo son 1. Cualquier otro valor se calcula sumando los dos valores contiguos de la fila de arriba.

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

Escribir la función recursiva  $\text{pascal}(n, k)$  que calcula el valor que se encuentra en la fila  $n$  y la columna  $k$ . Ejemplo:  $\text{pascal}(4, 2) \Rightarrow 6$