

### Recursión

- 1) Realice la función de fibonacci (pract 6 - ej 2) en forma recursiva
- 2) Escribir una función recursiva que devuelva la suma de los primeros N enteros
- 3) Programe un método recursivo que calcule el producto de un arreglo de números enteros. Para esto imagine, por ej. Vector V de tamaño 4, el producto de todos sus valores es igual a  $V[1] * (V[2] * (V[3] * V[4]))$
- 4) Escriba una función recursiva que ordene de menor a mayor un arreglo de enteros basándose en la siguiente idea: coloque el elemento más pequeño en la primera ubicación, y luego ordene el resto del arreglo con una llamada recursiva.
- 5) Implemente una función recursiva que nos diga si una cadena es palíndromo. Para esto tome el ejemplo propuesto en la diapositiva y resuelva el bug de los espacios en blanco

### Cadenas

- 6) Crea un programa que pida su nombre al usuario y lo escriba al revés (de la última letra a la primera: a partir de "Nacho" escribiría "ohcaN").
- 7) Crea un programa que pida su nombre al usuario y lo escriba alternando letras mayúsculas y minúsculas (por ejemplo, "nAcho" se mostraría como "NaChO").
- 8) Crea un programa que pida su nombre al usuario y diga cuántas vocales contiene (por ejemplo, "Aurora" tiene 4 vocales).
- 9) Crea un programa que pida su nombre al usuario y diga qué vocales contiene (en orden y sin repetir: por ejemplo, para "Aurora" deberá responder "aou").
- 10) Crea un programa que pida una frase al usuario y diga cuántas palabras contiene (pista: puedes contar los espacios, prestando atención en que no estén repetidos).
- 11) Crea un programa que pida al usuario su nombre y apellidos y los muestre con las mayúsculas y minúsculas correctas (aparecerán en mayúsculas la primera letra y la que haya tras cada espacio; las demás aparecerán en minúsculas. Por ejemplo, si introduce "nAcho cabaNes", es escribirá "Nacho Cabanes"). Utiliza la función de concatenación. Utilizar subprogramas para no repetir código.