
PRÁCTICA IV

Ejercicio 4.1

Crear una función recursiva que retorne el producto de dos números enteros en función de la suma.

Ejercicio 4.2

Crear una función recursiva que retorne el cociente de dos números enteros en función de la resta.

Ejercicio 4.3

Crear una función recursiva que invierta un arreglo unidimensional dinámico de enteros tal que su prototipo sea

```
void invertir (int *, unsigned int);
```

y cuyos parámetros son el arreglo a invertir y la cantidad de valores que hay en el mismo.

Ejemplo:

```
/* Valores originales: 6,2,7,1,8 */
invertir(a, 5);
/* El arreglo a ahora contiene: 8, 1, 7, 2, 6 */
```

Ejercicio 4.4

- Programar una función recursiva que retorne el valor máximo de un arreglo unidimensional dinámico números de enteros. El 0 (cero) marca el final de los valores útiles del arreglo.
- Programar una función recursiva que retorne la dirección de memoria del valor máximo de un arreglo unidimensional dinámico números de enteros. El 0 (cero) marca el final de los valores útiles del arreglo.

Ejercicio 4.5

- Programar una función recursiva que retorne el valor mínimo de un arreglo unidimensional dinámico números de enteros. El 0 (cero) marca el final de los valores útiles del arreglo.
- Programar una función recursiva que retorne la dirección de memoria del valor mínimo de un arreglo unidimensional dinámico números de enteros. El 0 (cero) marca el final de los valores útiles del arreglo.

Ejercicio 4.6

Programar una función recursiva que ordene en forma creciente un arreglo dinámico unidimensional de números enteros. El 0 (cero) marca el final de los valores útiles del arreglo. Utilizar la función recursiva programada para retornar la posición de memoria del valor mínimo del arreglo.

Ejercicio 4.7

Programar una función recursiva que reciba como parámetro una longitud (número entero positivo) e imprima todas las posibles combinaciones de letras minúsculas de dicha longitud. Por ejemplo:

```
imprimir_combinacion(5);
```

debería imprimir lo siguiente:

```
aaaaa
aaaab
aaaac
...
zzzzy
zzzzz
```

Ejercicio 4.8

- Programar una función recursiva para realizar la potenciación de números enteros (n^m) en función de la multiplicación.
- Programar la misma función pero intentando optimizarla de modo de minimizar la cantidad de multiplicaciones totales realizadas. Por ejemplo x^8 puede calcularse como $x^4 * x^4$ (calculando 1 vez x^4 y multiplicándolo por sí mismo). Luego x^4 puede calcularse como $x^2 * x^2$ y así sucesivamente.

Ejercicio 4.9

Se tiene una matriz de char N x M (constantes de cualquier valor entero positivo) donde pueden haber espacios (que representan lugares que se pueden llenar) o 'X' que representan y delimitan lugares que no se pueden llenar. Implementar una función recursiva que, dadas las coordenadas de un casillero, rellene los elementos permitidos de la matriz con el carácter '#'. Ejemplo:

Matriz inicial:

	X				
	X				
		X			X
			X	X	

Fila: 0

Columna: 0

Matriz rellena desde 0, 0:

#	X				
#	X				
#	#	X			X
#	#	#	X	X	#
#	#	#	#	#	#

Ejercicio 4.10

Alterar la función realizada en el ejercicio anterior, para que además de rellenar según el criterio del ejemplo anterior, permita propagar el llenado en diagonal. Ejemplo:

Matriz inicial:

				X	
X	X	X	X	X	
		X			X
			X	X	

Fila: 0

Columna: 2

Matriz rellena desde 2, 0:

				X	#
X	X	X	X	X	#
#	#	X	#	#	X
#	#	#	X	X	#
#	#	#	#	#	#

Ejercicio 4.11

Programar una función recursiva que reciba como parámetro un string conteniendo un texto ya cargado y que retorne la cantidad de vocales que hay en éste. El prototipo de la función debe ser:

```
unsigned int contar_vocales (char *);
```

Ejercicio 4.12

Programar una función recursiva que halle la salida de un laberinto de texto de 20 x 20 caracteres. La función debe recibir como parámetros el laberinto y las coordenadas de la entrada. A continuación, la función debe buscar recursivamente (probando todas las alternativas posibles) algún camino de salida dibujando el camino en el laberinto.

Ejercicio 4.13

Realizar una función booleana recursiva que reciba como parámetro un string conteniendo un texto ya cargado y que retorne si dicho texto es un palíndromo. Un texto es palíndromo cuando se lee igual de izquierda a derecha que de derecha a izquierda. Un ejemplo de texto palíndromo es “*A no dar a Maradona*”. No considerar diferentes a las letras mayúsculas de sus respectivas minúsculas. Ignorar los separadores.

Ejercicio 4.14

Programar una función booleana recursiva que reciba como parámetro de un arreglo unidimensional dinámico de números de enteros y retorne si dicho arreglo está ordenado en forma creciente. El 0 (cero) marca el final de los valores útiles del arreglo.