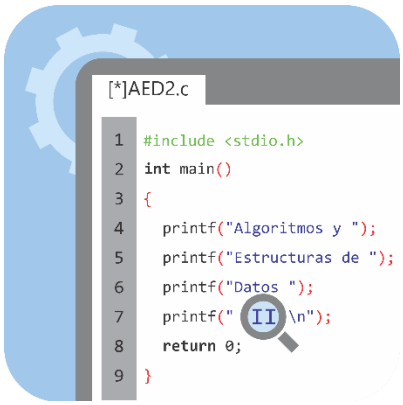

Algoritmos y Estructuras de Datos II

Práctico Nro 7
Recursividad

OBJETIVOS

- Conocer el concepto de *Recursividad*.
- Identificar las diferencias entre la implementación *iterativa* y la *recursiva*.
- Aprender a implementar *algoritmos recursivos*.

COMPETENCIAS

- Identificar, formular y resolver problemas mediante programación.
- Utilizar de manera efectiva técnicas y herramientas de aplicación para desarrollar software.
- Desempeñarse de manera efectiva en equipos de trabajo.
- Aprender en forma continua, autónoma y de manera colaborativa.

METODOLOGÍA

- El alumno deberá resolver individualmente los ejercicios propuestos.
- El alumno deberá codificar las soluciones en el lenguaje de programación C.
- Realizar consultas a través del canal de slack correspondiente a su comisión ó del aula virtual de la asignatura.

DURACIÓN

De acuerdo a la planificación de la asignatura, se deberán utilizar para la resolución de los ejercicios de esta serie, dos clases prácticas.

EJERCICIOS PROPUESTOS

Los siguientes ejercicios deberán resolverse utilizando funciones recursivas

1. Escribir un programa que permita mostrar por pantalla una cuenta regresiva, a partir de un valor ingresado por teclado. Programar una función recursiva que, al llegar la cuenta a cero, informe que el tiempo se ha agotado.

2. Escribir un programa que, a partir del ingreso de un número entero, y mediante la utilización de una función recursiva *verEnBinario*, permita visualizar el número binario equivalente por pantalla.

3. Escribir un programa que permita ingresar un número entero positivo y luego, mediante una función recursiva, muestre el número de forma invertida. Ej.: 123 - 321.

4. Escribir una función que calcule la división entera entre 2 números de manera recursiva. Recordar que la división matemática se define como una operación aritmética, que consiste en saber cuántas veces un número (divisor) está contenido en otra cifra (dividendo).

5. Modificar el programa codificado en el ejercicio 1 de esta serie, para que la cuenta regresiva se realice de a un segundo por vez.

> **Nota**
>
> La función **sleep** permite suspender la ejecución de un programa por un número de segundos determinados.
>
> Su prototipo es: void sleep(time_t seconds);
>
> Si el compilador no lo detecta automáticamente, agregar:
> Para Linux: #include <unistd.h>
> Para Windows: #include <windows.h>

6. Escribir un programa que, a partir del ingreso de dos números enteros positivos, calcule el producto de los mismos utilizando una función recursiva. Tener presente que la definición recursiva de la multiplicación de dos números a y b, se deriva de la definición de la multiplicación como una suma abreviada y la aplicación de la propiedad asociativa de la suma. La definición recursiva de la multiplicación es:

$$a * b = \begin{cases} a + (a * (b - 1)) & \text{si } b > 0 \\ 0 & \text{si } b = 0 \end{cases}$$

7. El siguiente pseudocódigo corresponde a una función que permite sumar los elementos de un vector en forma recursiva:

Función: sumaVec
Datos de entrada: un vector de números enteros, representado por la pareja de datos <*v*, *n*>. como precondición se considera que el vector ya tiene cargados un conjunto de valores válidos y que *n* es un valor comprendido entre el cero (el vector puede estar vacío) y NMAX (la dimensión máxima del array).
Datos de salida: la suma de los elementos del vector, es decir,
sumaVec(v, n) = ∑v[i], i = 0, ..., n

```
Función sumaVec ( v: tArray; n: Entero): Entero
Inicio
        Si (n = 0) Entonces
                { la solución para el caso base es directa }
                retorna 0;
        Sino
                { la solución para el caso general es recursiva }
                retorna sumaVec ( v, n-1 ) + v[n];
        Fin-Si
Fin
```

Desarrollar el código C correspondiente al pseudocódigo dado, tener presente al codificar el manejo del índice del arreglo.

8. Escribir un programa que permita el ingreso de dos valores enteros (la base entera y el exponente entero positivo), calcule la potencia y muestre los resultados por pantalla. Utilizar función recursiva.

   La definición recursiva de la operación exponenciación entera, es decir calcular la potencia de a^b, se deriva de la definición de la potencia como una multiplicación abreviada y la aplicación de la propiedad asociativa de la multiplicación. Entonces, la definición recursiva de exponenciación es:

$$a^b = \begin{cases} a * a^{b-1} & \text{si } b > 0 \\ 0 & \text{si } b = 0 \end{cases}$$

9. Escribir un programa que, mediante funciones, determine la suma de "N" números naturales. Mostrar la serie de los números desde 1 hasta N y el resultado de la suma.

10. Escribir un programa que permita ingresar una palabra y determinar si es palíndroma.

Nota

Un **palíndromo** es un término o una expresión que puede leerse tanto de izquierda a derecha como de derecha a izquierda (es decir, expresa lo mismo al ser leído de manera tradicional o al revés). Se trata del equivalente a lo que, respecto a los números, se conoce como capicúa.

11. Codificar un programa que, a partir del ingreso por teclado de un número entero positivo, permita calcular su factorial y visualizar el resultado por pantalla. Utilizar una función recursiva para el cálculo factorial.

$$\mathbf{n!} = \begin{cases} \mathbf{1} \rightarrow \text{Si } n = 0 \\ \mathbf{(n-1)! * n} \rightarrow \text{Si } n > 0 \end{cases}$$

12. Codificar un programa que, a partir del ingreso de un número entero positivo, calcule y muestre los números de la sucesión de Fibonacci. Utilizar una función recursiva.

$$\mathbf{F_0} = 0$$

$$\mathbf{F_1} = 1$$

$$\mathbf{F_n} = F_{(n-1)} + F_{(n-2)} \text{ , } n > 1$$