

Trabajo Práctico Nivel 8

Recursión con TD Arreglo de Registros: Concepto y Operaciones

Fecha de Inicio 29/10/2025 - Fecha de Fin 10/11/2025

Ejercicio 1: Dado el módulo iterativo que calcula el factorial de un número natural, se solicita replantearlo recursivamente y realizar la traza con num=6.

```
int factorial(int num){ int i,fact=1; for
    (i=num; i>=1; i--) fact=fact*i;
    return fact; }
```

Ejercicio 2: Para las siguientes situaciones, realice un módulo recursivo y la traza con los datos indicados para cada una:

- Dado un dígito D, determinar si pertenece a un número entero positivo N. Realizar la traza para D=1 y X= 45673.
- Dado un número natural, indicar si el mismo es perfecto, abundante o deficiente.

Nota: Un número natural X se dice que es deficiente si la suma de sus divisores (sin contarse a sí mismo) es menor a X. Si la suma de sus divisores es mayor a X se dice que el número es abundante y si es igual a X se dice que es perfecto.

Ejercicio 3: Realice la traza del siguiente programa con los valores 30 y 525. Elabore una consigna indicando qué hace el programa. #include <stdio.h> void averigua(int, int); int main(void) { int x;

```
printf("\n Ingrese un número natural ");
scanf("%d",&x); averigua(x,2);
return 0;
}
void averigua(int N, int f){
    if (N>1) if (N%f==0) {
        printf("\t %d",f);
        averigua(N/f, f);
    }
    else averigua(N, f+1); else
    printf("\t %d", 1);
}
```

Ejercicio 4:

Dada la siguiente definición recursiva:

$$\text{resto}(X, Y) \begin{cases} X & ; \text{ si } X < Y \\ \text{resto}(X-Y, Y) & ; \text{ si } X \geq Y \end{cases}$$

Formule el módulo recursivo y la traza para el cálculo del resto (22,5).

Nota: Esta función permite encontrar el resto de la división entera entre X e Y.

Ejercicio 5: Mostrar los dígitos de un número natural X de derecha a izquierda. Realizar la traza para X= 45673. Modificar el módulo para que muestre los dígitos de izquierda a derecha. (¿El módulo recursivo lleva return?)

Ejercicio 6: La siguiente función retorna el enésimo elemento de la sucesión de Fibonacci, realice el árbol de llamadas para n=4 y para n=7. Pruebe la función fibo, con n= 10, 20, 30, 40, 44, 45, 46, 47, 48, 49, 50 y 51. ¿Qué observa? ¿Por qué?

Ejercicio 7: Dada una lista de N números enteros que se ingresa ordenada, se pide crear un programa recursivo que permita: cargar y mostrar la lista, a continuación, buscar si encuentra un elemento ingresado por teclado, en caso de encontrarlo debe mostrar su posición y luego eliminarlo. *Nota: Hacer búsqueda binaria.*

Ejercicio 8: Se pide crear un programa recursivo que permita generar aleatoriamente una lista de N números enteros en el intervalo [A, B], luego presentar un menú de opciones (el cual debe ser recursivo) que permita realizar las siguientes tareas: mostrar la lista, Insertar un nuevo elemento en una posición ingresada por teclado, buscar un elemento en la lista.

Ejercicio 9: Se pide crear un programa recursivo que permita generar aleatoriamente una lista de N números enteros en el intervalo [A, B], luego presentar un menú de opciones (el cual debe ser recursivo) que permita realizar las siguientes tareas: ordenar ascendentemente utilizando el método Q-Sort, buscar un elemento en la lista (dicho elemento es ingresado por teclado).

Ejercicio 10: Se desea crear una lista de N nombres. Utilizar leeCad Recursivo A continuación, se desea ordenarla ascendentemente utilizando el método M-Sort y por último mostrar la lista resultante *Nota: TAD con todos sus módulos recursivos.*

Ejercicio 11: Diseñe un programa recursivo que permita, mediante un menú, administrar una lista de clientes. Los datos a registrar por cada cliente serán: Apellido y Nombre, DNI, saldo de cuenta. El programa debe además generar una lista de Deudores con los Clientes que posean un saldo de cuenta negativo y mostrar ordenado (Merge-Sort) la lista de Deudores, de acuerdo al saldo. *Nota: Ejercicio subido en el sitio para completar los módulos recursivos*

Ejercicio 12: Diseñe un programa recursivo que permita mediante un menú, administrar una lista de pacientes. Los datos a registrar por cada paciente serán: Apellido y Nombre, DNI, Tipo de Atención (Clínica, Odontología, Traumatología), Fecha de Atención. El programa debe generar, dado un Tipo de Atención y una Fecha de Atención, una nueva lista ordenada (Q-Sort) por el Apellido de los pacientes, que se deben atender ese día. También dadas dos fechas indicar la cantidad de pacientes que se han atendido entre esas fechas.

Nota: Ejercicio subido en el sitio para completar los módulos recursivos