**Conceptos de Recursividad**

1. **Introducción**

La recursividad es una implementación que permite resolver problemas de forma lógica, simplificada y efectiva, su práctica está inmersa en matemáticas, estructura de datos y más.

1. **Definición de Recursividad**

La recursión es un proceso en el que una entidad se define en función de sí misma. Aunque la recursión es poderosa, también puede consumir muchos recursos, por lo que es crucial analizar cuándo y cómo aplicarla. Aunque un problema pueda ser recursivo por definición, no siempre es el método de solución más adecuado. Antes de implementar un proceso recursivo en aplicaciones prácticas, es necesario demostrar que el nivel máximo de recursión es finito y pequeño, ya que cada llamada recursiva consume memoria para almacenar el estado del proceso. En términos de programación, una función es recursiva cuando se llama a sí misma.

1. **Importancia de la Recursividad**

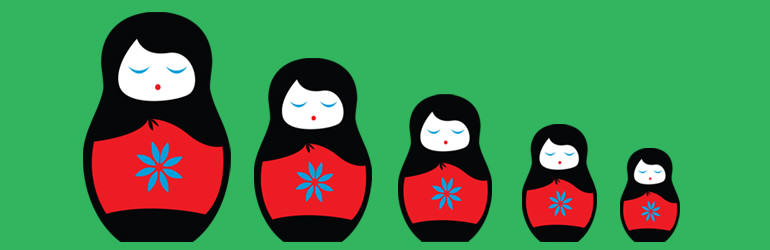
Permite trabajar de manera simplificada, ya que su naturaleza consiste en convertir a un problema en subproblemas, usando, la menor capacidad posible de espacio, un tiempo rápido de ejecución y un entendimiento mejor de la función.

1. **Uso de la Recursividad**

Cómo se ha mencionado, aunque la recursividad es poderosa, depende totalmente del problema.

Para usar este proceso, se debe,

* Parametrizar las variables que se usarán durante cada llamada.
* Plantear el caso base, mismo que permitirá acabar otra llamada y no caer en un bucle infinito.
* Realizar los cambios respectivos que se harán en las variables de las llamadas.
* Colocar donde se necesiten las llamadas de la misma función.



1. **Implementación de la Recursividad**

Main.cpp

#include <iostream>

#include "windows.h"

#include "validaciones.h"

#include "math.h"

**int** **calcular\_potencia**(**int**, **int**);

**int** **invertir\_numero**(**int**, **int**);

**int** **calcular\_potencia**(**int** base, **int** power){

**if**(power != **0**){

**return** base\*calcular\_potencia(base,power-**1**);

    }

**else**{

**return** **1**;

    }

}

**int** **invertir\_numero**(**int** numero, **int** resultado){

**if**(numero > **0**){

        resultado = resultado\***10** + numero%**10**;

        invertir\_numero(numero/**10**, resultado);

    }

**else**{

**return** resultado;

    }

}

**void** **mostrar\_menu\_principal**(**int** opcion){

    system("cls");

    std::cout << "**\n**--------------RECURSIVIDAD-----------------**\n**";

    std::cout << "**\n**Seleccione una opcion:**\n\n**";

    std::cout << (opcion == **1** ? "> " : "  ") << "Calcule la potencia de un numero entero positivo**\n**";

    std::cout << (opcion == **2** ? "> " : "  ") << "Invertir un numero entero positivo**\n**";

    std::cout << (opcion == **3** ? "> " : "  ") << "Salir**\n**";

}

**int** **main**()

{

**int** numero1, numero2;

**int** opcion = **1**;

**char** tecla;

    HANDLE consoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE);

    CONSOLE\_CURSOR\_INFO cursorInfo;

    GetConsoleCursorInfo(consoleHandle, &cursorInfo);

    cursorInfo.bVisible = false;

    SetConsoleCursorInfo(consoleHandle, &cursorInfo);

**do**{

        mostrar\_menu\_principal(opcion);

        tecla = \_getch();

**switch**(tecla){

**case** **72**: // Flecha arriba

**if** (opcion > **1**) opcion--;

**break**;

**case** **80**: // Flecha abajo

**if** (opcion < **3**) opcion++;

**break**;

**case** **13**: // Enter

**if**(opcion == **1**){

                    cursorInfo.bVisible = true;

                    SetConsoleCursorInfo(consoleHandle, &cursorInfo);

                    numero1 = ingresar\_enteros("Ingrese la base:");

                    printf("**\n**");

                    numero2 = ingresar\_enteros("Ingrese el exponente:");

                    printf("**\n**");

                    printf("%d ^ %d = %d**\n**", numero1, numero2, calcular\_potencia(numero1, numero2));

system("pause");

cursorInfo.bVisible = false;

                    SetConsoleCursorInfo(consoleHandle, &cursorInfo);

                }

**else** **if**(opcion == **2**){

                    cursorInfo.bVisible = true;

                    SetConsoleCursorInfo(consoleHandle, &cursorInfo);

                    numero1 = ingresar\_enteros("Ingrese el numero el cual quiere invertir:");

                    printf("**\n**");

                    numero2 = **0**;

                    printf("Numero Invertido = %d**\n**", invertir\_numero(numero1, numero2));

system("pause");

cursorInfo.bVisible = false;

                    SetConsoleCursorInfo(consoleHandle, &cursorInfo);

                }

**else** **if**(opcion == **3**){

                    system("cls");

    std::cout << "Saliendo del programa." << std::endl;

    exit(**0**);

                }

**break**;

**default:**

**break**;

        }

    }**while**(true);

    system("pause");

cursorInfo.bVisible = true;

    SetConsoleCursorInfo(consoleHandle, &cursorInfo);

}

1. **Referencias**

Bhadaniya, S. (2023, May 29). *Recursion in C++: Types, Examples & Advantages*. FavTutor. https://favtutor.com/blogs/recursion-cpp

Ceballos, F. (2020) C/C++ Curso de programación. 5ª. Edición.