Explicación códigos ALSE programas en C++

David Eduardo Martínez Gutiérrez

***Ingeniería Electrónica***

***1. Suma de elementos de un arreglo***

Este programa solicita al usuario que ingrese 5 números enteros. Dichos números   
se almacenan en un arreglo y posteriormente se muestran en pantalla junto con   
la suma total de todos ellos.

***Código***:

#include <iostream>  
using namespace std;  
  
int main(){  
   int arreglos[5];  
   int suma = 0;  
  
   cout << "Ingrese los números que desea" << endl;  
   for(int i = 0; i < 5; i++){  
       cout << "Digite el dato " << (i+1) << ": ";  
       cin >> arreglos[i];  
   }  
  
   cout << "Los números dentro del arreglo son:" << endl;  
   for(int i = 0; i < 5; i++){  
       cout << "Número " << (i+1) << ": " << arreglos[i] << endl;  
       suma += arreglos[i];  
   }  
  
   cout << "La suma de los datos es: " << suma << endl;  
  
   return 0;  
}

***Explicación detallada:***

- Declaración de variables:  
 Se crea un arreglo `arreglos[5]` para almacenar 5 enteros, y una variable `suma`   
 inicializada en 0.  
  
- Entrada de datos:  
 Mediante un ciclo `for`, el programa pide 5 veces un número al usuario,   
 utilizando `cin` para guardarlos en el arreglo.  
  
- Recorrido y suma:  
 Otro ciclo `for` recorre los elementos del arreglo, imprime cada número y   
 los va acumulando en la variable `suma`.  
  
- Salida final:  
 Se muestra en pantalla la suma de todos los elementos ingresados.  
  
- Conclusión:  
 El programa ilustra cómo manejar arreglos, ciclos `for`, entrada/salida con `cin` y `cout`,   
 y cómo realizar operaciones acumulativas (suma en este caso).

***2. Cálculo de interés compuesto***

Este programa calcula el interés compuesto aplicando la fórmula matemática:  
  
   M = C \* (1 + r/n)^(n\*t)  
  
Donde:  
- M = monto final acumulado  
- C= capital inicial  
- r = tasa de interés anual (en forma decimal, ej: 0.05 equivale a 5%)  
- n = número de veces que se aplica el interés en un año  
- t = cantidad de años

***Código:***

#include <iostream>  
using namespace std;  
  
double potencia(double base, int exponente) {  
   double resultado = 1;  
   for (int i = 0; i < exponente; i++) {  
       resultado \*= base;  
   }  
   return resultado;  
}  
  
double interes\_compuesto(double capital, double tasa, int n, int t){  
   int exponente = n \* t;  
   return capital \* potencia(1 + tasa / n, exponente);  
}  
  
int main(){  
   double capital, tasa;  
   int n, t;  
  
   cout << "Digite su capital: ";  
   cin >> capital;  
  
   cout << "Digite la tasa de interés anual (en decimal, ej: 0.05): ";  
   cin >> tasa;  
  
   cout << "Digite el número de veces que se aplica en un año: ";  
   cin >> n;  
  
   cout << "Digite el número de años: ";  
   cin >> t;  
  
   double total = interes\_compuesto(capital, tasa, n, t);  
   double ganancia = total - capital;  
  
   cout << "El monto total con interés compuesto es: " << total << endl;  
   cout << "La ganancia fue de: " << ganancia << endl;  
  
   return 0;  
}

***Explicación detallada:***

- Función `potencia`:  
 Implementa un algoritmo iterativo para calcular una potencia (base^exponente),   
 multiplicando repetidamente la base.  
  
- Función `interes\_compuesto`:  
 Aplica la fórmula matemática del interés compuesto. Calcula el exponente `n\*t`,   
 que corresponde al número total de veces que se aplicará el interés durante todo el periodo.  
  
- Entrada de datos:  
 El programa solicita al usuario: capital inicial, tasa de interés anual,   
 número de aplicaciones en un año (`n`) y años totales (`t`).  
  
- Cálculos:  
 Se obtiene el monto final (`total`) y se calcula la ganancia como   
 `ganancia = total - capital`.  
  
- Salida de resultados:  
 Se imprime el monto final con interés compuesto y la ganancia neta.  
  
- Conclusión:  
 Este programa combina programación estructurada (funciones) con   
 conceptos matemáticos (potencia y fórmulas financieras).

***3. Máximo Común Divisor (MCD)***

Este programa determina el Máximo Común Divisor (MCD)entre dos números   
usando el Algoritmo de Euclides, que es un método eficiente y clásico   
para este cálculo.

***Código:***

#include <iostream>  
using namespace std;  
  
int MCD(int n1, int n2){  
   while(n2 != 0){  
       int residuo = n1 % n2;  
       n1 = n2;  
       n2 = residuo;  
   }  
   return n1;  
}  
  
int main(){  
   int n1, n2;  
   cout << "Digite un número: ";  
   cin >> n1;  
   cout << "Digite otro número: ";  
   cin >> n2;  
  
   cout << "El MCD de los números es: " << MCD(n1, n2) << endl;  
   return 0;  
}

***Explicación detallada:***

- Función `MCD`:  
 Implementa el Algoritmo de Euclides:  
 1. Se calcula el residuo de la división entre `n1` y `n2`.  
 2. `n1` toma el valor de `n2` y `n2` toma el valor del residuo.  
 3. Este proceso se repite hasta que `n2` sea 0.  
 4. Cuando ocurre, `n1` contiene el MCD.  
  
- Entrada de datos:  
 El usuario ingresa dos números enteros.  
  
- Proceso y salida:  
 El programa invoca la función `MCD` y muestra en pantalla el resultado.  
  
- Ejemplo:  
 Si el usuario ingresa 48 y 18:  
 - 48 % 18 = 12 → (n1=18, n2=12)  
 - 18 % 12 = 6 → (n1=12, n2=6)  
 - 12 % 6 = 0 → (n1=6, n2=0)  
 - El MCD es 6.  
  
- Conclusión:  
 Este programa es un ejemplo clásico de uso de funciones, ciclos `while` y   
 operadores aritméticos para resolver un problema matemático real.

Link GitHub:

Codigos 1,2 y 3

<https://github.com/davidedumg-hub/codigoparcial.cpp.git>