



Tarea: Arreglos unidimensionales y funciones en C++

Objetivo general:

Aplicar el uso de arreglos unidimensionales junto con diferentes tipos de funciones (void, por referencia, por valor, y con retorno) y aplicar un método de ordenamiento (Burbuja) utilizando C++.

Cálculo del precio final de un producto

Enunciado 0: Una tienda aplica descuentos a un producto antes de venderlo al cliente. Al precio base se le aplican dos tipos de descuentos:

- Descuento por promoción 15%
- Descuento por cliente frecuente 10%

Elabora un programa en C++ que calcule el precio final después de aplicar ambos descuentos.

Requerimientos del programa

1. Crear la función double calcularDescuento(double precio, double porcentaje);
 - Recibe el precio base y un porcentaje de descuento.
 - Devuelve el valor descontado.
2. Crear la función double calcularPrecioFinal(double precioBase, double porcPromo, double porcCliente);
 - Llama a calcularDescuento para obtener los dos descuentos.
 - Retorna el precio final después de restarlos.
 - Los valores de descuento deben estar establecidos dentro del cálculo.
3. En la función main:
 - Solicitar al usuario el precio base del producto.
 - Solicitar los porcentajes de descuento por promoción y por cliente frecuente.
 - Llamar a la función calcularPrecioFinal.
 - Mostrar en pantalla:
 - ➔ Descuento por promoción
 - ➔ Descuento por cliente frecuente
 - ➔ Precio final del producto



```
//Descuentos para un producto de una tienda con sus dos tipos de descuentos por promoción o frecuencia.
#include <iostream>
using namespace std;

void CalculoDescuento(double precio){
    double porcentaje1, porcentaje2;
    string frecuencia, promocion;

    porcentaje1=(0.10 * precio);
    porcentaje2=(0.15 * precio);

    cout<<" Ingrese si es un cliente frecuente y ademas si el producto tiene promocion" << endl;
    cin>>frecuencia>>promocion;

    if(frecuencia=="si"){
        cout<<"Se le aplica un descuento por cliente frecuente del "<< "10%"<< endl;
        cout<<"Su valor aplicado del descuento es de $"<< porcentaje1 << endl;
    }
    else{
        cout<<"No se le aplica ningun descuento de cliente preferencial y su valor es de: $"<< precio<< endl;
    }
    if(promocion=="si"){
        cout<<"Se le aplica un descuento de promocion del "<< " 15% " << endl;
        cout<<"Su valor aplicado por el descuento de la promocion es: $"<<porcentaje2<< endl;
    }
    else{
        cout<<"No se le aplica el descuento de la promocion su valor se mantiene en : $"<< precio<< endl;
    }
    cout<<("-----")<< endl;
}

void Calculoprefinal(double precio){
    double porcentaje1,porcentaje2;

    porcentaje1=(0.10 * precio);
    porcentaje2=(0.15 * precio);

    double DescPromo=porcentaje1;
    double DescCliente=porcentaje2;

    double PrecioFinal=precio-(DescPromo+DescCliente);
    cout << "Precio base: $" << precio << endl;
    cout << "Descuento por promocion: $" << DescPromo << endl;
    cout << "Descuento por cliente frecuente: $" << DescCliente << endl;
    cout << "Precio final del producto: $" << PrecioFinal << endl;
}

int main (){
    double precio;
    cout<<" Ingrese el precio del producto escogido $";
    cin>>precio;
    CalculoDescuento(precio);
    Calculoprefinal(precio);

    return 0;
}
```

Funciones con void (sin retorno)

Enunciado 1: Diseñe un programa en C++ que permita ingresar cinco números enteros en un arreglo unidimensional y luego, mediante una función void, muestre los valores almacenados en pantalla.

Enunciado 2: Cree un programa que solicite al usuario ingresar las calificaciones de 4 estudiantes, las almacene en un arreglo y utilice una función void para mostrar todas las notas registradas.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SOFTWARE



```
//Diseño de dos programas con un arreglo pwo que estos ademas almacenen entre 4 o 5 numeros, ya sea para notas o una Lista de numeros pare.
#include <iostream>
using namespace std;
void arreglo(int arr[], int tam){
cout<< "----- ENUNCIADO 1 -----" << endl;
    cout << "Los numeros ingresados son: ";
    for (int i = 0; i < tam; i++) {
        cout << arr[i] << " ";
    }
    cout<<endl;
}
void calificaciones(int cal[], int est){
cout<< "----- ENUNCIADO 2 -----" << endl;
    cout << "Las calificaciones de los cuatro estudiantes es: ";
    for(int a=0;a<est;a++){
        cout<< cal[a] << " ";
    }
}
int main(){
    int numeros[5];
    cout << "Ingrese 5 numeros enteros:" << endl;
    for (int i = 0; i < 5; i++) {
        cin >> numeros[i];
    }
    arreglo(numeros, 5);
    cout<< "----- FIN DEL ENUNCIADO 1 -----" << endl;

    int notas[4];
    cout<< "Ingrese las calificaciones de los cuatro estudiantes: " << endl;
    for(int a=0;a<4;a++){
        cin >> notas[a];
    }
    calificaciones(notas, 4);
    cout<< endl;
    cout<< "----- FIN DEL ENUNCIADO 2 -----" << endl;

    return 0;
}
```

Funciones con parámetros por referencia

Enunciado 3: Desarrolle un programa que lea los elementos de un arreglo de 5 números enteros y, mediante una función que use paso por referencia, calcule la suma total de los elementos.

Elegir eliminar un dato según el ingreso por teclado



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SOFTWARE



```
/*Crecion de un arreglo de 5 numeros aplicando una funcion de referencia y tambien calculando la suma de sus elementos.
//Arreglo que permite el almacenamiento de varios numeros mediante una funcion de referencia.
#include <iostream>
using namespace std;

void calcularSuma(int arr[], int tan, int &sum) {
    suma = 0;
    for (int i = 0; i < tan; i++) {
        ...
    }
}

void eliminarDato(int arr[], int &tan, int valor) {
    bool encontrado = false;
    for (int i = 0; i < tan; i++) {
        if (arr[i] == valor) {
            for (int j = i; j < tan - 1; j++) {
                arr[j] = arr[j + 1];
            }
            tan--;
            encontrado = true;
            break;
        }
    }
    if (!encontrado) {
        cout << "El valor " << valor << " no se encuentra en el arreglo." << endl;
    }
}

void mostrarArreglo(int arr[], int tan) {
    cout << "Arreglo actual:" << endl;
    for (int i = 0; i < tan; i++) {
        ...
    }
    cout << endl;
}

int main() {
    const int MAX = 5;
    int numeros[MAX];
    int tan = MAX;
    int suma;
    cout << "Ingrese 5 numeros enteros:" << endl;
    for (int i = 0; i < MAX; i++) {
        ...
        cin >> numeros[i];
    }
    calcularSuma(numeros, tan, suma);
    cout << "La suma total es: " << suma << endl;

    int valorEliminar;
    cout << "Ingrese el valor que desea eliminar:" << endl;
    cin >> valorEliminar;
    eliminarDato(numeros, tan, valorEliminar);
    mostrarArreglo(numeros, tan);

    calcularSuma(numeros, tan, suma);
    cout << "La nueva suma total es: " << suma << endl;
}

return 0;
}
```

Enunciado 4: Implemente un programa que permita almacenar varios números en un arreglo e identifique, mediante una función que use parámetros por referencia, cuál es el valor mayor del arreglo.

```
/*Funciones dentro de un almacenamiento de numeros en un
arreglo obteniendo el valor mas alto dentro de este mismo*/
#include <iostream>
using namespace std;

void encontrarMayor(int arr[], int x, int &mayor) {
    mayor = arr[0];
    for (int i = 1; i < x; i++) {
        if (arr[i] > mayor) {
            mayor = arr[i];
        }
    }
}

int main() {
    const int Can = 5;
    int numeros[Can];
    int mayor;

    cout << "Ingrese " << Can << " numeros enteros:" << endl;
    for (int i = 0; i < Can; i++) {
        ...
        cin >> numeros[i];
    }

    encontrarMayor(numeros, Can, mayor);

    cout << "El valor mayor del arreglo es: " << mayor << endl;
}

return 0;
}
```



Funciones con parámetros por valor

Enunciado 5: Escriba un programa que almacene varios números en un arreglo y, mediante una función con parámetros por valor, multiplique cada elemento por un número ingresado por el usuario, mostrando el resultado en pantalla sin alterar el arreglo original.

```
/* Creacion de un arreglo obteniendo el resultado sin mostrar
ningun cambio en el arreglo original.*/
#include <iostream>
using namespace std;

void multiplicarArreglo(int arr[], int tam, int multiplicador) {
    cout << "Resultados de la multiplicacion: ";
    for (int i = 0; i < tam; i++) {
        cout << arr[i] * multiplicador << " ";
    }
    cout << endl;
}

int main() {
    int numeros[5];
    int multiplicador;

    cout << "Ingrese 5 numeros enteros:" << endl;
    for (int i = 0; i < 5; i++) {
        cin >> numeros[i];
    }

    cout << "Arreglo original: ";
    for (int i = 0; i < 5; i++) {
        cout << numeros[i] << " ";
    }
    cout << endl;

    cout << "Ingrese el numero multiplicador: ";
    cin >> multiplicador;

    multiplicarArreglo(numeros, 5, multiplicador);

    cout << "Arreglo original despues de la funcion: ";
    for (int i = 0; i < 5; i++) {
        cout << numeros[i] << " ";
    }
    cout << endl;
}

return 0;
}
```

Enunciado 6: Cree un programa que permita almacenar números enteros en un arreglo e implemente una función con paso por valor que cuente cuántos de esos números son pares. La función debe retornar la cantidad encontrada.

```
/*Arreglo que permite el almacenamiento de varios valores enteros e implementando
una función para la obtención de numeros pares*/
#include <iostream>
using namespace std;

int contarPares(int tab[], int can) {
    int contador = 0;
    for (int i = 0; i < can; i++) {
        if (tab[i] % 2 == 0) {
            contador++;
        }
    }
    return contador;
}

int main() {
    int numeros[5];

    cout << "Ingrese 5 numeros enteros:" << endl;
    for (int i = 0; i < 5; i++) {
        cin >> numeros[i];
    }

    int cantidadPares = contarPares(numeros, 5);

    cout << "La cantidad de numeros pares es: " << cantidadPares << endl;
}

return 0;
}
```



Ejercicio con return (función que devuelve un valor)

Enunciado 8: Realice un programa que calcule y retorne el promedio de los elementos de un arreglo (función con return). La función deberá recibir el arreglo y su tamaño, calcular el promedio y devolverlo como double. En el programa principal se mostrará el promedio con dos decimales.

```
/*Aplicacion de notas que nos permita el retorno del promedio en los elementos dentro de un arreglo*/
#include <iostream>
#include <iomanip>
using namespace std;

double calcularPromedio(int list[], int tama) {
    int suma = 0;
    for (int i = 0; i < tama; i++) {
        suma += list[i];
    }
    return (double)suma / tama;
}

int main() {
    int numeros[5];

    cout << "Ingrese 5 numeros enteros:" << endl;
    for (int i = 0; i < 5; i++) {
        cin >> numeros[i];
    }

    double promedio = calcularPromedio(numeros, 5);

    cout << endl;
    cout << "El promedio de los elementos es: " << promedio << endl;
}

return 0;
```

Factura con descuentos (condicionales)

Enunciado 9: Desarrolle un programa en C++ que lea: nombre del producto, precio unitario y cantidad. Aplique descuentos según la cantidad:

- Si la cantidad es **2**, aplicar **10%** de descuento sobre subtotal.
- Si la cantidad es **3 o más**, aplicar **15%** de descuento sobre subtotal.
Además, si el subtotal (antes de descuentos) supera 200, aplicar un **descuento adicional de 5%** (acumulativo).

Implemente funciones para: cargar datos (void), calcular subtotal y descuentos (double con return), aplicar descuentos por referencia (modificando el total), y mostrar la factura (void). Mostrar subtotal, descuentos aplicados y total final.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SOFTWARE



```
/* Creacion de una factura con descuento aplicando diferentes descuento*/
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

void cargarDatos(string &nombre, double &precioUnitario, int &cantidad) {
    cout << "Ingrese nombre del producto: ";
    cin >> nombre;
    cout << "Ingrese precio unitario: ";
    cin >> precioUnitario;
    cout << "Ingrese cantidad: ";
    cin >> cantidad;
}

double calcularSubtotal(double precioUnitario, int cantidad) {
    return precioUnitario * cantidad;
}

void aplicarDescuentos(double subtotal, int cantidad, double &total, double &descuentoTotal) {
    descuentoTotal = 0;

    if (cantidad == 2) {
        descuentoTotal += subtotal * 0.10;
    } else if (cantidad >= 3) {
        descuentoTotal += subtotal * 0.15;
    }

    if (subtotal > 200) {
        descuentoTotal += subtotal * 0.05;
    }

    total = subtotal - descuentoTotal;
}

void mostrarFactura(string nombre, double precioUnitario, int cantidad, double subtotal, double descuentoTotal, double total) {
    cout << "----- FACTURA -----" << endl;
    cout << "Producto: " << nombre << endl;
    cout << "Precio unitario: $" << precioUnitario << endl;
    cout << "Cantidad: " << cantidad << endl;
    cout << "Subtotal: $" << subtotal << endl;
    cout << "Descuentos aplicados: $" << descuentoTotal << endl;
    cout << "Total final: $" << total << endl;
}

int main() {
    string nombre;
    double precioUnitario, subtotal, total, descuentoTotal;
    int cantidad;

    cargarDatos(nombre, precioUnitario, cantidad);
    subtotal = calcularSubtotal(precioUnitario, cantidad);
    aplicarDescuentos(subtotal, cantidad, total, descuentoTotal);
    mostrarFactura(nombre, precioUnitario, cantidad, subtotal, descuentoTotal, total);

    return 0;
}
```

Tarjeta Metro/Bus Quito (recarga y pagos)

Enunciado 10: Crea un programa en C++ que simule una tarjeta de transporte del Metro/Bus de Quito.

El programa debe permitir **recargar saldo, pagar pasajes y mostrar el saldo disponible después de cada pasada**.

La tarjeta debe tener un **saldo inicial de 0**.

Debe permitir **recargar** un monto (por ejemplo, \$5, \$10, etc.).

Cada vez que el usuario pague un pasaje:

- Se **descuenta** el valor del pasaje (por ejemplo, \$0.45).
- Se **muestra el saldo actual** después del pago.

El programa debe llevar un **contador de pasadas** (viajes pagados).



Usar **funciones** para organizar el código:

- void recargar(double &saldo, double monto)
- bool pagar(double &saldo, double tarifa)
- void mostrarEstado(double saldo, int pasadas)

===== TARJETA METRO/BUS QUITO =====

Saldo actual: \$0.00

Ingresé monto a recargar: 5

Recarga exitosa. Saldo actual: \$5.00

Pagando pasaje de \$0.45 ...

Pago exitoso. Saldo actual: \$4.55

Pagando pasaje de \$0.45 ...

Pago exitoso. Saldo actual: \$4.10

Pasadas realizadas: 2

```
/*Simulación de varias funciones para el cálculo de recarga de una tarjeta y además
el cálculo de cobre de dos viajes.*/
#include <iostream>
#include <iomanip>
using namespace std;

void recargar(double &saldo, double monto) {
    saldo += monto;
    cout << "Recarga exitosa. Saldo actual: $" << saldo << endl;
}

bool pagar(double &saldo, double tarifa) {
    if (saldo >= tarifa) {
        saldo -= tarifa;
        cout << "Pago exitoso. Saldo actual: $" << saldo << endl;
        return true;
    } else {
        cout << "Saldo insuficiente. No se pudo pagar el pasaje." << endl;
        return false;
    }
}

void mostrarEstado(double saldo, int pasadas) {
    cout << "===== TARJETA METRO/BUS QUITO =====" << endl;
    cout << "Pasadas realizadas: " << pasadas << endl;
    cout << "Saldo final: $" << fixed << setprecision(2) << saldo << endl;
    cout << "===== TARJETA METRO/BUS QUITO =====" << endl;
}

int main() {
    double saldo = 0.0;
    double monto;
    double tarifa = 0.45;
    int pasadas = 0;

    cout << "===== TARJETA METRO/BUS QUITO =====" << endl;
    cout << "Saldo actual: $" << saldo << endl;

    cout << "Ingresé monto a recargar: ";
    cin >> monto;
    recargar(saldo, monto);

    cout << "Pagando pasaje de $" << tarifa << " ..." << endl;
    if (pagar(saldo, tarifa)) pasadas++;

    cout << "Pagando pasaje de $" << tarifa << " ..." << endl;
    if (pagar(saldo, tarifa)) pasadas++;

    mostrarEstado(saldo, pasadas);

    return 0;
}
```



Saldo final: \$4.10

Lineamientos de entrega

1. El trabajo debe desarrollarse en C++ utilizando funciones para cada ejercicio.
2. Cada programa debe incluir:
 - Nombre del estudiante.
 - Descripción breve del ejercicio.
 - Comentarios dentro del código.
3. En github crear un **repositorio** y suba los archivos .cpp y el documento .docx con los códigos correctamente indentados.
4. Poner el enlace de github en un pdf.