



EXAMEN BIMESTRAL PROGRAMACION

INDICACIONES

- **Material permitido:** Solo se puede usar el IDE de C++. No se permite calculadora, celular ni dispositivos electrónicos.
- **Silencio absoluto:** Durante el examen no se permite conversar ni pedir materiales a otros compañeros.
- **Tiempo asignado:** el examen tiene una duración de 160 minutos. Una vez finalizado el tiempo, se recibirán todos los archivos.cpp en el aula virtual, sin extensiones de tiempo.
- **Prevención de copia:** Cualquier intento de copiar, compartir respuestas o usar materiales no autorizados será motivo de anulación inmediata del examen.

DESARROLLO

EJERCICIO 1

Dado el siguiente arreglo:

```
string provincias[24]={"Pichincha"}
```

Se pide implementar las siguientes funciones(trabajar con datos quemados, no es necesario pedir datos por consola):

- a) Crear una función que permita recorrer y mostrar el arreglo de provincias
- b) Crear una función que permita agregar una nueva provincia al arreglo existente.
- c) Crear una función que permita cambiar el nombre de una provincia existente, identificándola por su índice en el arreglo
- d) Crear un arreglo para eliminar una provincia simulando la eliminación de la misma al reemplazar su nombre por una cadena vacía ("")



```
#include <iostream>
#include <string>
using namespace std;

string provincias[25] = {
    "Azuay", "Bolívar", "Cañar", "Carchi", "Chimborazo", "Cotopaxi",
    "El Oro", "Esmeraldas", "Galápagos", "Guayas", "Imbabura", "Loja",
    "Los Ríos", "Manabí", "Morona Santiago", "Napo", "Orellana", "Pastaza",
    "Pichincha", "Santa Elena", "Santo Domingo de los Tsáchilas",
    "Sucumbíos", "Tungurahua", "Zamora Chinchipe", " "
}
int cantidad = 25;

void mostrarProvincias() {
    cout << "Listado de provincias:" << endl;
    for (int i = 0; i < cantidad; i++) {
        if (provincias[i] != "") {
            cout << i << " - " << provincias[i] << endl;
        }
    }
}

void agregarProvincia(string nueva) {
    if (cantidad < 25) {
        provincias[cantidad] = nueva;
        cantidad++;
    } else {
        cout << "No se pueden agregar más provincias (arreglo lleno)." << endl;
    }
}

void cambiarProvincia(int indice, string nuevoNombre) {
    if (indice >= 0 && indice < cantidad) {
        provincias[indice] = nuevoNombre;
    } else {
        cout << "Índice inválido." << endl;
    }
}

void eliminarProvincia(int indice) {
    if (indice >= 0 && indice < cantidad) {
        provincias[indice] = "";
    } else {
        cout << "Índice inválido." << endl;
    }
}
```



```
C:\Users\User\Downloads> cor
Listado de provincias:
0 -> Azuay
1 -> Bolívar
2 -> Cañar
3 -> Carchi
4 -> Chimborazo
5 -> Cotopaxi
6 -> El Oro
7 -> Esmeraldas
8 -> Galápagos
9 -> Guayas
10 -> Imbabura
11 -> Loja
12 -> Los Ríos
13 -> Manabí
14 -> Morona Santiago
15 -> Napo
16 -> Orellana
17 -> Pastaza
18 -> Pichincha
19 -> Santa Elena
20 -> Santo Domingo de los Tsáchilas
21 -> Sucumbíos
22 -> Tungurahua
23 -> Zamora Chinchipe
24 ->
Listado de provincias:
0 -> Azuay
1 -> Manabí
2 -> Cañar
```

Nota: No es necesario realizar un menú, directamente invocar a las funciones en el main

EJERCICIO2

Dados los siguientes arreglos en C++:

```
string nombres []={"Juan Garcia","Luis Coba","Ana Solis","Andy Salas"}
string cargos []={"director","subdirector","analista","analista"}
int ids []={104,103,102,101}
double sueldosBase []={2400,1560,1250,1250}
```

- Crear una función void que recorra el arreglo de empleados y muestre en pantalla todos sus datos: nombre, cargo, id y sueldo base.
- Crear una función que devuelva double para calcular y retorne el **promedio salarial** de los empleados.
- Crear una función que devuelva el sueldo más alto que existe.
- Crear una función calcularSalarioNeto que en base al sueldoBase calcule el **salario neto** de cada empleado y **retorne** un arreglo de doubles con el nuevo salario.



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SOFTWARE



- a. El salario neto se obtiene aplicando dos descuentos al sueldo base:
 - i. Seguridad social: 9.45%
 - ii. Impuestos: 7.65%
- b. Genere un nuevo arreglo con los sueldos netos y compruebe su funcionamiento utilizando la función del inciso (a).

```
#include <iostream>
#include <string>
using namespace std;

string nombres[] = {"Juan Garcia", "Luis Coba", "Ana Solis", "Andy Salas"};
string cargos[] = {"director", "subdirector", "analista", "analista"};
int ids[] = {104, 103, 102, 101};
double sueldosBase[] = {2400, 1560, 1250, 1250};
int cantidad = 4;

void mostrarEmpleados(double sueldos[]) {
    cout << "Listado de empleados:" << endl;
    for (int i = 0; i < cantidad; i++) {
        cout << "Nombre: " << nombres[i]
            << " | Cargo: " << cargos[i]
            << " | ID: " << ids[i]
            << " | Sueldo: " << sueldos[i] << endl;
    }
}

double promedioSalarial() {
    double suma = 0;
    for (int i = 0; i < cantidad; i++) {
        suma += sueldosBase[i];
    }
    return suma / cantidad;
}

double sueldoMaximo() {
    double max = sueldosBase[0];
    for (int i = 1; i < cantidad; i++) {
        if (sueldosBase[i] > max) {
            max = sueldosBase[i];
        }
    }
    return max;
}

double* salarioNeto() {
    static double sueldosNetos[];
    for (int i = 0; i < cantidad; i++) {
        double descuentoSS = sueldosBase[i] * 0.0945;
        double descuentoImp = sueldosBase[i] * 0.0765;
        sueldosNetos[i] = sueldosBase[i] - descuentoSS - descuentoImp;
    }
    return sueldosNetos;
}
```

```
== Empleados con sueldo base ==
Listado de empleados:
Nombre: Juan Garcia | Cargo: director | ID: 104 | Sueldo: 2400
Nombre: Luis Coba | Cargo: subdirector | ID: 103 | Sueldo: 1560
Nombre: Ana Solis | Cargo: analista | ID: 102 | Sueldo: 1250
Nombre: Andy Salas | Cargo: analista | ID: 101 | Sueldo: 1250
Promedio salarial: 1615
Sueldo més alto: 2400
== Empleados con sueldo neto ==
Listado de empleados:
Nombre: Juan Garcia | Cargo: director | ID: 104 | Sueldo: 1989.6
Nombre: Luis Coba | Cargo: subdirector | ID: 103 | Sueldo: 1293.24
Nombre: Ana Solis | Cargo: analista | ID: 102 | Sueldo: 1036.25
Nombre: Andy Salas | Cargo: analista | ID: 101 | Sueldo: 1036.25
-----
Process exited after 0.2594 seconds with return value 0
Presione una tecla para continuar . . .
```

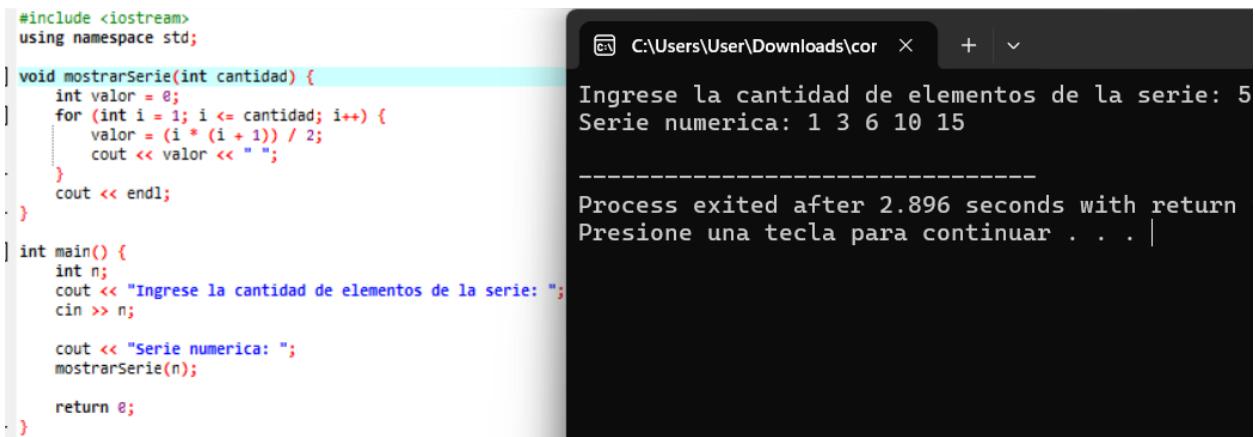
Nota: No es necesario realizar un menú, directamente invocar a las funciones en el main

EJERCICIO 3

Identifica el patrón y crea una función básica (la estudiante determinada si es void o con retorno y sus parámetros), a fin de generar la siguiente serie numérica:

1 3 6 10 15 21 28 36 etc...

El usuario ingresa por teclado la cantidad de elementos de la serie numérica a mostrar.



```
#include <iostream>
using namespace std;

void mostrarSerie(int cantidad) {
    int valor = 0;
    for (int i = 1; i <= cantidad; i++) {
        valor = (i * (i + 1)) / 2;
        cout << valor << " ";
    }
    cout << endl;
}

int main() {
    int n;
    cout << "Ingrese la cantidad de elementos de la serie: ";
    cin >> n;

    cout << "Serie numerica: ";
    mostrarSerie(n);

    return 0;
}
```

C:\Users\User\Downloads\cor x + ▾
Ingrese la cantidad de elementos de la serie: 5
Serie numerica: 1 3 6 10 15

Process exited after 2.896 seconds with return
Presione una tecla para continuar . . . |

EJERCICIO 4

El Metro de Quito necesita un sistema automatizado para sus tarjetas ciudadanas. Desarrollar un programa en C++ que simule la generación y recarga de **tarjetas**, utilizando **al menos 3 funciones con retorno** y **al menos 2 funciones void con paso por referencia obligatorio**. Al menos una función debe ser **dependiente** (llamar a otra función internamente).

Información del Sistema:

Tarifas por Viaje para registro de usuario:

- **Estudiante:** \$0.22 (Para validar se requiere confirmar si cuenta con carnet estudiantil)
- **General:** \$0.45
- **Adultos Mayores (65+):** \$0.22
- **Preferencial (con discapacidad):** \$0.10 (Para validar se requiere confirmar si cuenta con carnet CONADIS)

Tarjeta Recargable (tarjeta ciudadana):

- **Costo inicial único:** \$2.00 + IVA 15% (no incluye saldo inicial)
- **Recarga inicial obligatoria:** \$3.00 (para activar tarjeta)
- **Recarga posterior mínima:** \$0.50 en adelante (hasta 100\$ como máximo)

Requisitos Funcionales del Sistema:

El programa debe permitir:

1. Registrar Usuario
 - a. Pedir cedula, nombre(solo el primer nombre), edad
 - b. Seleccionar tipo de tarifa
 - c. Validar edad y confirmación de documentos según tarifa elegida



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SOFTWARE



- d. Calcular costo de tarjeta nueva con IVA 15% y mostrar el saldo inicial(por defecto inicia con \$3.00 y se asume que el usuario paga este valor)
2. Procesar recarga de tarjeta ciudadana
 - a. Pedir la cedula del usuario y validar si existe
 - b. Recargar la tarjeta del usuario, sumando el valor recargado al monto original
3. Pagar pasaje (de acuerdo con las tarifas anteriormente configurada)
 - a. Pagar siempre y cuando haya saldo en la tarjeta
 - b. Calcular el saldo restante
4. Mostrar usuarios registrados
 - a. Recorrer los arreglos de cedula, nombres, tarifa, saldo y mostrar los datos guardados
5. Salir de la aplicación.

Validaciones obligatorias:

- Estudiante: edad 5 hasta 18 años + confirmación mediante carnet estudiantil
- Tercera edad: mayor a 65 años + confirmación mediante cedula
- Preferencial: debe validarse que tenga carnet CONADIS

Restricciones Técnicas:

Cantidad de Funciones sugeridas:

- **Mínimo 6 funciones** en total (el estudiante puede agregar más, a consideración de su lógica).
- **Al menos 2 funciones void** (con paso por referencia obligatorio)
- **Al menos 3 funciones con retorno** (return)
- **Al menos 1 función dependiente** (una función que llame a otra internamente)
- **Arreglos unidimensionales(tamaño 5 mínimo)** uno para almacenar nombres, otro para cédulas, otro para la tarifa y finalmente uno para el saldo

Posibles funciones void:

- Función para asignar tarifa según opción del menú
- Función para procesar recarga y calcular saldo

Posibles funciones return:

- Función para validar edad según tipo de tarifa → retorna bool
- Función para calcular costos→ retorna double
- Función para calcular tarifa según el tipo de usuario -> retorna double
- Función para si el usuario cumple con los requisitos de edad y documentación->retorna bool

Nota: Los estudiantes son libre de crear las funciones adicionales que consideren necesarias siempre que cumplas las restricciones técnicas(2 void con paso por referencia, 3 con retorno, 1 dependiente y 3 arreglos).



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE SOFTWARE



```
#include <iostream>
#include <string>
using namespace std;

string cedula, nombre;
int edad, tipotarifa;
double saldo = 0;

double calcularCostoTarjeta() {
    return 0 + (0 * 0.15);
}

double obtenerTarifaviaje(int tipo) {
    if (tipo == 1) return 0.22;
    if (tipo == 2) return 0.45;
    if (tipo == 3) return 0.22;
    if (tipo == 4) return 0.18;
    return 0.45;
}

bool validarTarifa(int tipo, int edad, bool carnetEst, bool carnetConadis) {
    if (tipo == 1) return (edad >= 5 && edad <= 18 && carnetEst);
    if (tipo == 2) return true;
    if (tipo == 3) return (edad >= 65);
    if (tipo == 4) return carnetConadis;
    return false;
}

void registrarUsuario() {
    cout << "*** REGISTRO DE USUARIO ***" << endl;
    cout << "Ingrese cedula: ";
    cin >> cedula;
    cout << "Ingrese nombre: ";
    cin >> nombre;
    cout << "Ingrese edad: ";
    cin >> edad;

    cout << "Seleccione tipo de tarifa:" << endl;
    cout << "1. Estudiante ($0.22)" << endl;
    cout << "2. General ($0.45)" << endl;
    cout << "3. Adulto Mayor ($0.22)" << endl;
    cout << "4. Preferencial ($0.10)" << endl;
    cout << "Opcion: ";
    cin >> tipoTarifa;

    bool carnetEst = false, carnetConadis = false;
    if (tipotarifa == 1) {
        cout << "(Tiene carnet estudiantil? (1=sí, 0=No): ";
        int op; cin >> op; carnetEst = (op == 1);
    }
}
```

Seleccione una opcion: 2
*** RECARGA DE TARJETA ***
Ingrese numero de cedula: 1722023031
Usuario: mary
Saldo actual: \$0
Ingrese monto a recargar (mínimo \$1.00, máximo \$100): \$10
Recarga exitosa. Saldo actual: \$10
== MENU METRO DE QUITO ==
1. Registrar Usuario
2. Recargar Tarjeta
3. Pagar Pasaje
4. Salir
Seleccione una opcion: 3
*** PAGAR PASAJE ***
Ingrese número de cedula: 1722023031
Usuario: mary
Saldo actual: \$10
Costo de pasaje: \$0.45
Pasaje pagado exitosamente.
Saldo restante: \$9.55
== MENU METRO DE QUITO ==
1. Registrar Usuario
2. Recargar Tarjeta
3. Pagar Pasaje
4. Salir
Seleccione una opcion: 3
*** PAGAR PASAJE ***
Ingrese número de cedula: 1722023031
Usuario: mary

Salidas (sugeridas) por Pantalla, el estudiante completa las opciones en base a la lógica que analice para cumplir los requerimientos.

(A continuación, se presenta una guía sugerida)

INICIO
<pre>===== SISTEMA METRO DE QUITO GESTION DE TARJETAS CIUDADANAS ===== 1. Registrar nuevo usuario 2. Recargar tarjeta 3. Pagar pasaje 4. Mostrar usuarios registrados 5. Salir Seleccione una opcion:1</pre>

Luego se requiere ir configurando cada opción (guía sugerida)

REGISTRO DE USUARIOS



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
DESARROLLO DE SOFTWARE



== REGISTRO DE USUARIO ==

Ingrese cedula:1104136625

Ingrese nombre:Juan

Ingrese edad:33

Seleccione tipo de tarifa:

1. Estudiante (\$0.22)
2. General (\$0.45)
3. Adulto Mayor (\$0.22)
4. Preferencial (\$0.10)

Opcion:2

Costo de tarjeta ciudadana (incluye IVA): \$2.3

Tarifa asignada: \$0.45 por viaje

Recarga inicial obligatoria: \$3

Saldo actual: \$3

Usuario registrado exitosamente.

RECARGAS

== RECARGA DE TARJETA ==

Ingrese numero de cedula:1104136625

Usuario: Juan

Saldo actual: \$3

Ingrese monto a recargar (minimo \$0.5, maximo \$100): \$10

Recarga exitosa. Saldo actual: \$13

PAGAR PASAJE

== PAGAR PASAJE ==

Ingrese numero de cedula:1104136625

Usuario: Juan

Saldo actual: \$13

Costo del pasaje: \$0.45

Pasaje pagado exitosamente.

Saldo restante: \$12.55



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
DESARROLLO DE SOFTWARE



MOSTRAR USUARIOS REGISTRADOS

== LISTA DE USUARIOS ==

Cedula: 1104136625 | Nombre: Juan | Tarifa: \$0.45 | Saldo: \$12.55

Rúbrica

Ejercicio	Porcentaje de Calificación
Ejercicio 1	25%
Ejercicio 2	30%
Ejercicio 3	15%
Ejercicio 4	30%