

Trabajo Práctico N°3

1)

- a) Escribir la función **es_par(numero)** que determine si el número entero que recibe a través del parámetro “numero” es o no es par. La función deberá retornar 1 si “numero” efectivamente es par y 0 en caso contrario.
- b) Escribir un programa que solicite al usuario el ingreso de un número entero, y que utilizando la función, **es_par(numero)** del ejercicio anterior, determine si el número ingresado es par o impar, imprimiendo un mensaje que lo informe al usuario por pantalla.

2)

- a) Implementar la función **suma_digitos(numero)** que, dado un número entero como argumento, retorne la suma de sus dígitos. Por ejemplo, si el argumento de la función es 438, deberá retornar 15 (pues $4+3+8=15$).
- b) Escribir un programa que solicite al usuario el ingreso de un número entero, y que utilizando la función del ejercicio anterior, muestre en pantalla un mensaje que informe al usuario la suma de los dígitos del número ingresado.

3)

- a) Implementar la función **mostrar_suma_digitos(numero)** que, dado un número entero como argumento, imprima en pantalla la suma de sus dígitos. ¿Qué valor retorna esta función?
- b) Escribir un programa que solicite al usuario el ingreso de un número entero, y que utilizando la función del ejercicio anterior, muestre en pantalla un mensaje que informe al usuario la suma de los dígitos del número ingresado.

4)

- a) Implementar la función **promedio_de_edades(edad1, edad2, edad3)**.
- b) Escribir un programa que solicite el ingreso por teclado de tres edades y luego llame a la función creada en el punto anterior. Finalmente, informar el resultado en pantalla.

5)

- a) Implementar la función **cantidad_vocales(cadena)**, la cual debe informar la cantidad de vocales sin tilde que contiene la cadena pasada por parámetro,

independientemente si son minúsculas o mayúsculas.

- b) Escribir un programa que solicite el ingreso por teclado de una cadena y luego llame a la función creada en el punto anterior. Finalmente, informar el resultado en pantalla.

6)

- a) Implementá la función `maximo(num_1, num_2)` que calcule y retorne el máximo entre `num_1` y `num_2`, ambos positivos.
- b) Luego, usá esa función para implementar un programa que solicite la lectura de 5 números que sean positivos, en caso de ingresar uno negativo volver a solicitar un nuevo número, e informe el máximo número leído.

7) Escribir un programa que imprima el menú mostrado a continuación y solicite al usuario elegir una opción, la cual se debe ejecutar para luego volver a mostrar el menú repetidas veces, hasta que el usuario decida salir:

'A': Invertir número.

'B': Sumatoria de dígitos.

'C': Decir si son múltiplos.

'D': Salir.

Dependiendo del carácter ingresado por el usuario, realizar la acción indicada en el menú: A: leer un número e invertir sus dígitos para luego imprimirla en orden inverso; B: solicitar un número e imprimir la suma de sus dígitos; C: solicitar el ingreso de dos números y decir si el primero es múltiplo del segundo; D: terminar el programa.

Si el usuario ingresa un carácter que no es la opción D, se debe ejecutar lo que corresponda y volver a mostrar el menú para permitirle elegir otra opción. Si el carácter ingresado por el usuario no es A, B, C ni D, mostrarle un mensaje de error y continuar mostrándole el menú y solicitando una opción hasta que ingrese una que sea válida.

Implementar las funciones que considere necesarias.

8)

- a) Implementar la función `invertir_cadena(cadena)` que muestre una cadena pasada por parámetro en orden inverso.

- b) Escribir un programa que solicite el ingreso por teclado de una cadena y luego llame a la función creada en el punto anterior para informar el resultado en pantalla.
- 9)
a) Implementar la función mayuscula(cadena) que transforme una cadena pasada por parámetro en mayúsculas.
b) Escribir un programa que solicite el ingreso por teclado de una cadena y luego llame a la función creada en el punto anterior para informar el resultado en pantalla.
- 10) Escribí un programa que solicite el ingreso de números enteros, finalizando luego de leer un número para el cual la suma de sus dígitos sea menor que 10 o mayor que 50. Informar la cantidad de números impares leídos. Utilizá las funciones es_par(número) y suma_dígitos(número) implementadas anteriormente.
- 11) Sin modificar el código existente, completar las líneas punteadas en cada uno de los programas mostrados a continuación. Suponer que en cada uno se han importado los encabezados necesarios y se usa el espacio de nombres estándar:

PROGRAMA 1)

```
.....mayusculas(string cadena) {  
    for (int i = 0; i < cadena.length(); i++) {  
        cadena[i] = toupper(cadena[i]);  
    }  
.....  
  
int main() {  
    string nombre;  
    cout << "Cuál es tu nombre?: ";  
    getline(cin>>ws, nombre);  
.....mayusculas(nombre);  
    cout << "Hola, " << nombre << "!" << endl;  
    return 0;  
}
```

INTRODUCCIÓN A LA PROGRAMACIÓN IMPERATIVA

PROGRAMA 2)

```
.....mayusculas(string ..... cadena) {  
    for (int i=0; i<cadena.length(); i++){  
        cadena[i] = toupper(cadena[i]);  
    }  
}  
  
int main() {  
    string nombre;  
    cout << "Cuál es tu nombre?: ";  
    getline(cin>>ws, nombre);  
    mayusculas(nombre);  
    cout << "Hola, " << nombre << "!" << endl;  
    return 0;  
}
```

PROGRAMA 3)

```
..... validarEleccion() {  
    char opcion;  
    do{  
        cout << "Desea confirmar el préstamo? Ingrese S ó N: ";  
        cin>>opcion;  
        opcion = tolower(opcion);  
    } while (opcion != 's' and opcion != 'n');  
    return .....;  
}  
  
int main() {  
    cout << " *** BIENVENIDO AL SISTEMA DEL BANCO MONEY MONEY *** " << endl;  
    cout << "Ingrese monto para obtener un préstamo: ";  
    float monto;  
    cin >> monto;  
    cout << "El interés es del 25%." << endl;  
    if (validarEleccion() == 's')  
        cout << "Ahora usted nos debe " << monto + (monto*0.25) << endl;  
    else  
        cout << "Lamentamos que no quiera hacer negocios con nosotros.";  
    return 0;  
}
```

INTRODUCCIÓN A LA PROGRAMACIÓN IMPERATIVA

- 12) Sin utilizar la computadora, indicar qué valores se imprimirían en cada caso. Luego verificar compilando y ejecutando cada uno.

CASO 1)

```
void primeraFuncion(int x) {
    x = x+8;
    cout << "Valor de x luego de la primera función:" << x << endl;
}

void segundaFuncion(int & x) {
    x = x+5;
    primeraFuncion(x);
    cout << "Valor de x luego de la segunda función:" << x << endl;
}

int main() {
    int x = 2;
    segundaFuncion(x);
    return 0;
}
```

CASO 2)

```
void funcion(int a, int & y, int & z) {
    int b = 1;
    y = y+1;
    z = z+a;
    b = 4;
}

int main() {
    int a = 2;
    int b = 3;
    funcion(a+b, a, a);
    cout << "Variable a:" << a << endl;
    cout << "Variable b:" << b << endl;
    return 0;
}
```

¿Qué cambiaría si en la función se reemplazara el nombre del parámetro **a** por el nombre **x**? Es decir:

```
void funcion(int x, int & y, int & z) {
    int b = 1;
    y = y+1;
    z = z+x;
    b = 4;
}
```

CASO 3)

```
int sumatoriaPares(int numero) {
    int suma = 0;
    while (numero != 0) {
        if ((numero%10)%2 == 0) {
            suma += numero%10;
        }
        numero = numero/10;
    }
    return suma;
}

void alterarNumero(int a, int & b, int c) {
    a = sumatoriaPares(c);
    b = a;
}

int main() {
    int i = 4;
    int numero = 90210;
    alterarNumero(i, numero, sumatoriaPares(numero));
    cout << "Número: " << numero << endl;
    cout << "Variable i: " << i << endl;
    return 0;
}
```

INTRODUCCIÓN A LA PROGRAMACIÓN IMPERATIVA

- 13) ¿Qué salida produce el siguiente fragmento de código? Si genera un error, especificar cuál es y por qué se produce. Proponer posibles soluciones.

```
#include <iostream>
using namespace std;
string x = "azul";

void primera() {
    cout<<"Me gusta el color "<<x<<endl;
}

void segunda() {
    string y = "verde";
    cout<<"Me gusta el color " <<y<<endl;
}

int main()
{
    primera();
    segunda();
    cout<<"Mis colores favoritos son "<<x<<" y "<<y<<endl;
}
```

- 14) Escribe un programa en C++ que simule la gestión de una cuenta bancaria. El programa debe permitir al usuario realizar las siguientes operaciones:

- Ingresar un monto a la cuenta.
- Retirar un monto de la cuenta.
- Mostrar el saldo actual de la cuenta.

Requisitos:

- Utiliza una variable global para almacenar el saldo total de la cuenta.
- Implementa funciones con variables locales para realizar cada una de las operaciones mencionadas.
- Asegúrate de manejar adecuadamente los casos donde no hay fondos suficientes para realizar un retiro.

Ejemplo:

Saldo Inicial: \$1000

1. Ingresar un monto a la cuenta.
2. Retirar un monto de la cuenta.
3. Mostrar el saldo actual de la cuenta.
0. Salir

Ingresá tu elección: 1

Ingresá el monto a ingresar: 500

Monto ingresado con éxito.

Ingresá tu elección: 2

Ingresá el monto a retirar: 800

Fondos insuficientes. Saldo actual: \$500.

Ingresá tu elección: 3

Saldo actual: \$500.

Ingresá tu elección: 0

¡Adiós!

- 15) Identificar los errores, si los hay, de las siguientes declaraciones de arreglos. Para aquellos que son correctos especificar: la cantidad de elementos ocupados y los valores almacenados.

```
int a[4] = {8, 7, 6, 4, 3};  
int b[] = {8, 7, 6, 4};  
const int MAX = 4;  
char c[MAX];  
long d[5];
```

¿Qué salida produce el siguiente código?

```
int i, arreglo[10];
for (i = 0; i < 10; i++)
    arreglo[i] = 2*i;
for (i = 0; i < 10; i++)
    cout << arreglo[i] << " ";
    cout << endl;
for (i = 0; i <= 10; i = i + 2)
    cout << arreglo[i] << " ";
```

- 16) Escribir un programa que solicite cinco números enteros, los almacene en un arreglo y luego calcule y muestre en pantalla el promedio de esos números.
- 17) Escribir una función que reciba un arreglo de números enteros y devuelva la suma de los elementos que son pares.
- 18) Escribir una función que reciba un arreglo de enteros y devuelva el promedio de los elementos que se encuentren en posiciones impares del arreglo.
- 19) Ordenamiento de números:
 - a) Desarrolla una función que ordena en forma ascendente los elementos de un arreglo de 10 números enteros.
Por ejemplo, el arreglo con los elementos [2,5,1,4,4,8,12,9,7,11] quedaría ordenado como [1,2,4,4,5,7,8,9,11,12]
 - b) ¿Qué debería modificar del algoritmo anterior para que los ordene en orden descendente?
- 20) Eliminar duplicados: crear una función que elimine los elementos duplicados de un arreglo de N números enteros, dejando una única ocurrencia de cada número.