Práctica 1

Temario:

- Variables
- Tipos de datos simples
- Cadenas de caracteres
- Entrada / salida básica
- Estructuras de control
- Funciones y parámetros

Ejercicios:

Ejercicio 1.

Escribir un programa que imprima los números del 1 al 100, mostrando de a cinco por renglón.

Ejercicio 2.

Contar la cantidad de números negativos leídos desde teclado. La serie termina con el valor cero (0), el cual no debe procesarse. ¿Qué cambiaría si debiera procesarse también el último número ingresado (0)?

Ejercicio 3.

Leer tres letras desde teclado e indicar cuál de ellas viene primero en el alfabeto.

Ejercicio 4.

Escribir un programa que, dado un número (entero largo) expresado en segundos, informe el equivalente en horas, minutos y segundos.

Ejercicio 5.

Escribir un programa que lea una fecha en el formato numérico AAAAMMDD y la informe con el formato DD/MM/AAAA.

Precondición: los valores ingresados corresponden a una fecha válida.

Ejercicio 6.

Escribir un programa que calcule la suma de una determinada cantidad de elementos de la serie armónica: 1/1 + 1/2 + 1/3 + … + 1/n, donde **n** se especifica por el usuario.

Ejercicio 7.

Escribir un programa que visualice el más grande, el más pequeño y el promedio de N números enteros. El valor de N se solicita al usuario al comienzo del programa y luego se le permite introducir los números.

Ejercicio 8.

Leer una secuencia de caracteres que finaliza con la letra minúscula 'n', la cual debe ser procesada. Contabilizar la cantidad de cada una de las letras vocales minúsculas. También informar la cantidad de caracteres leídos en total y el porcentaje de letras vocales minúsculas encontradas en la secuencia.

Ejercicio 9.

Escribir un bucle para validar lo ingresado por el usuario: se debe solicitar el ingreso de un carácter y sólo se debe interrumpir el bucle cuando el usuario haya ingresado el carácter 's' o el carácter 'n'. La solicitud de ingreso del dato debe hacerse al menos una vez antes de evaluar si el bucle debe interrumpirse o no. No utilizar "while true" como condición. A continuación, si el usuario ingresó la letra 's', imprimir "usted eligió SÍ" y, si ingresó la 'n' imprimir "usted eligió NO".

Nota: para evaluar la condición de corte del bucle considerando mayúsculas y minúsculas es posible utilizar las funciones *toupper(carácter)* y *tolower(carácter)*, que convierten el carácter pasado por parámetro a mayúscula o a minúscula respectivamente (si el carácter no es una letra, no se modifica).

Ejercicio 10.

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 imprimirlo 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.

Ejercicio 11.

Leer una cadena de caracteres (string) de teclado e imprimirla en orden inverso.

Ejercicio 12.

Leer una frase de teclado e informar la longitud de la última palabra. Se considera al espacio como separador de palabras.

Precondición: el string no finaliza con espacios.

Ejercicio 13.

Leer una cadena de caracteres de teclado e, independientemente de cómo ha sido ingresada, convertir la primera letra a mayúscula y el resto a minúsculas. Considerar que la cadena puede estar vacía.

Nota: la función toupper(carácter) convierte un carácter a mayúscula si éste es una letra de la A a la Z (si es cualquier otro carácter, incluso letras acentuadas, no hace nada). De la misma forma, tolower(carácter) convierte a minúscula.

Precondición: si la cadena no está vacía, el primer carácter es una letra.

Ejemplo: si se ingresa la cadena “EsTO ES UNa cadENA de Texto”, el resultado debería ser “Esto es una cadena de texto”

Ejercicio 14.

Implementar un programa que solicite una palabra y verifique si es palíndromo. Una palabra es palíndromo si puede leerse igual de izquierda a derecha que de derecha a izquierda.

Precondición: todos los caracteres son minúsculas y no se ingresarán vocales acentuadas ni diéresis.

Para los siguientes ejercicios, modularizar cuando corresponda:

Ejercicio 15.

Dado el siguiente programa cuya función main está completa, implementar las funciones que falten para que compile y realice las tres operaciones indicadas en el menú.

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

//Implementar aquí las funciones invocadas en main

int main() {
   int opcion;
   do {
       string cadena;
       cout << "Ingresar frase: ";
       getline(cin>>ws, cadena);

       cout << "MENÚ: " << endl;
       cout << "1. Cantidad total de vocales (mayúsculas y minúsculas)." << endl;
       cout << "2: Contar cuántas veces aparece un carácter." << endl;
       cout << "3: Mostrar caracteres en posiciones (índice) pares." << endl;
       cout << "0: Salir." << endl;
       cout << "Opción elegida: ";
       cin >> opcion;

       switch (opcion) {
       case 1:
           cout << "Cantidad de vocales: " << cantidadVocales(cadena) << endl;
           break;
       case 2:
           cout << "Ingresar carácter a contar: ";
           char caracter;
           cin >> caracter;
           cout << "Cantidad encontrada: " << contar(cadena, caracter) << endl;
           break;
       case 3:
           caracteresPares(cadena);
           cout << endl;
           break;
       case 0:
           break;
       default:
```

```
            cout << "Opción no válida" << endl;
            break;
        }
    } while (opcion != 0);
}
```

Ejercicio 16.

Escribir una función que reciba un punto de coordenadas (X,Y) desde el teclado y devuelva TRUE si el punto pertenece a la recta de ecuación Y = 3X + 2 ó FALSE en caso contrario.

Nota: sólo se debe hacer la función, aunque puede ser útil realizar la invocación y utilizar el valor de retorno para imprimir algo en pantalla, a fines de probar el algoritmo.

Ejercicio 17.

Escribir un programa que determine si un año es bisiesto.

Nota: un año es bisiesto si es múltiplo de 4. Los años múltiplos de 100 no son bisiestos, excepto que también sean múltiplos de 400.

Ejemplo: 2000 es bisiesto, 1800 no lo es.

Ejercicio 18.

Escribir un programa que, dado un año y una cantidad de días transcurridos desde el 1 de enero, muestre la fecha en formato dd/mm/aaaa. Validar que la cantidad de días dada sea válida (entre 1 y 365 ó entre 1 y 366). Para esto se debe considerar si el año es bisiesto o no.

Ejemplos: en el año 2019, el día número 256 corresponde al 13/09/2019.

En el año 2000 (bisiesto), el día número 60 corresponde a la fecha 29/02/2000.

En el año 2014, el día número 366 no es válido ya que 2014 tuvo 365 días.

Ejercicio 19.

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:

```
#include <iostream>
#include <string>
```

```
using namespace std;
```

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;
}
```

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: ";
        opcion = tolower(opcion);
```

```
    } while (opcion != 's' and opcion != 'n');
    return ……;
}
int main() {
    cout << " *** BIENVENIDO AL SISTEMA DEL BANCO MONEYMONEY *** " << 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;
}
```

Ejercicio 20.

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 << "Numero: " << numero << endl;
```

```
   cout << "Variable i: " << i << endl;
   return 0;
}
```

Ejercicio 21.

¿Qué errores tiene este fragmento de código y por qué?

```
void funcion(const int & x, int & y, int z) {
       x++;
       y = 12;
       z = 20;
}

int main() {
       int a = 0;
       int b = 9;
       funcion(a, 17, b);
       cout << "Variable a: " << a << endl;
       cout << "Variable b: " << b << endl;
       return 0;
}
```

Ejercicio 22.

Completar los parámetros y las instrucciones necesarias en el cuerpo de la siguiente función, de manera que la impresión final muestre los valores de las variables *a* y *b* intercambiados.

```
void intercambiarValores (……) {
    ……
    ……
    ……
}

int main() {
       int a = 10;
       int b = 25;
       intercambiarValores(a,b);
       cout << "Variable a: " << a << endl;
       cout << "Variable b: " << b << endl;
       return 0;
}
```

¿Podría hacerse esto mismo si ambas variables están pasadas por valor?