

Ejercicio 1

Escribir un único programa en C++ que contenga las siguientes funciones:

- `void mostrarMensaje ()`
(Imprime un mensaje de bienvenida.)
- `int leerPar ()`
(Devuelve un entero par leído por teclado.)
- `void calcularCubo (int num)`
(Imprime el cubo de un entero num.)

Ejercicio 2

Escribir un programa en C++ que contenga las siguientes funciones:

- `double promedio (int N)`
(Lee por teclado N números reales y devuelve su promedio.)
- `double absoluto (double val)`
(Devuelve el valor absoluto, sin usar la librería matemática.)
- `int techo (double val)`
(Imprime el valor entero mayor más cercano.)
- `int piso (double val)`
(Imprime el valor entero menor más cercano.)

Ejercicio 3

Escribir un programa en C++ que implemente el juego $Ta - Te - Ti$ basándose en el siguiente ejemplo:

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

/** Variables y constantes globales */
// Tablero de 3x3 (alcance global)
const int N = 3;
char tablero[N][N];

/** Prototipos de funciones */
/* Inicializa el tablero sin marcas -> caracter '-' */
void inicializar (void);
/* Imprime el tablero y sus marcas */
void imprimir (void);
/* Marca el caracter correspondiente (X u O)
en una posición libre que ingresa el jugador
por teclado */
void marcar (char );
/* Determina si el juego terminó en empate
(no hay más espacios por marcar) o
si hay un ganador.*/
bool fin ( char );
```

```

int main ()
{
    char jugador = 'X';

    inicializar();
    while ( !fin(jugador) )
    {
        // Alterno jugadores
        switch ( jugador)
        {
            case 'O':
                jugador = 'X';
                break;
            case 'X':
                jugador = 'O';
                break;
        }
        // Doy oportunidad para marcar
        marcar(jugador);
        // Muestro el estado del tablero
        imprimir();
    }

    system("PAUSE");
}

```

Ejercicio 4

Escribir un programa en C++ que implemente el juego Battalla Naval donde el programa posiciona aleatoriamente M barcos sobre un tablero de dimensión $N \times N$. Más de un barco puede asociarse a una misma posición. El usuario gana el juego cuando hunde todos los barcos. El programa debe basarse en el siguiente ejemplo:

```

#include<iostream>
#include<cstdlib> // Permite usar función rand() que devuelve un entero aleatorio.
using namespace std;

/** Variables y constantes globales **/
const int M = 10; // cantidad de naves
const int N = 5; // dimensión del tablero NxN
int tableroNaves[N][N]; // tablero donde se posicionan las M naves
int tableroHundidos[N][N]; // tablero donde se anotan las naves hundidas

/** Prototipos de Funciones **/
/* Inicializa ambos tableros y posiciona aleatoriamente las M naves */
void inicializar (void);
/* Muestra en pantalla el tablero con naves hundidas */
void imprimir (void);
/* Determina si se hunde barcos dada una coordenada x,y
ingresada por teclado. Actualiza el tableroHundidos en
caso afirmativo */
void disparar ( void );
/* Determina si se han hundidos todas las naves, es decir,
si tableroHundidos es igual al tableroNaves */
bool fin ( void );

```

```
int main ()
{
    inicializar();
    while ( !fin() )
    {
        disparar();
        imprimir();
    }
    cout << "Todas las naves han sido hundidas!!" << endl;
    // system("PAUSE");
}
```