Objetivo de la práctica:
- Practicar el uso de vectores y matrices. Aplicar los conceptos vistos con anterioridad principalmente referentes a las correcta modularización del programa

NOTA 1:Durante la práctica todos los ejercicios deberán ser guardados ***temporalmente*** *en el directorio \tmp. Una vez finalizada la misma y transferidos los ficheros a un disquete, se deberá eliminar dicho directorio.*

Conceptos básicos: vectores, matrices y array multidimensionales.

Un *array* es una secuencia de objetos del mismo tipo. Estos objetos se llaman elementos del *array* y se numeran consecutivamente 0, 1, 2, 3, ... El acceso a cada uno de los elementos de un vector se realiza mediante un índice.
El tipo de elementos almacenados en el array puede ser cualquier tipo de dato C++.
Adoptaremos el convenio de emplear **`typedef`** para crear este tipo de dato estructurado:
Cuando el array necesita un solo índice para ser definido se llama ***vector*** y lo declararemos de la siguiente forma***:***

```
const int MAX = 20;

typedef int Vector20[MAX]; //define un vector de 20 enteros

Vector20 mi_vector; //declaro un vector de 20 enteros.
```

```
Ejemplo de acceso a un elemento del vector:   mi_vector[3] = 45;
    //asigna el valor 45 a la cuarta componente
```

Cuando el array necesita 2 índices para ser definido se llama ***matriz*** y lo declararemos de la siguiente forma:

```
const int FIL = 10;
const int COL = 4;

typedef int Matriz10x4[FIL][COL];

Matriz10x4 mi_matriz;
```

```
Ejemplo de acceso a un elemento de la matriz: mi_matriz[1][3] = 32;
```

Cuando el array necesita más de 2 índices para ser definido se llama ***array multidimensional:***

```
const int FILA1 = 10;
...
const int FILAN =12;

typedef int Array_n_columnas[FILA1][FILA2]. . . .[FILAN];

Array_n_columnas mi_array_n_columnas;
```

```
Ejemplo de acceso a un elemento de la matriz:
    mi_array_n_columnas [1][2] . . .[10] = 20;
```

Los arrays **se pasan por defecto por referencia** como argumentos de una función. La única diferencia con los tipos simples es que no se usa él '&' ya que basta simplemente el nombre del array. Para pasar un array únicamente como parámetro de entrada se pone la palabra **const** delante del tipo del array.

PROBLEMAS

1.- Realizar un programa que lea de teclado 10 números enteros y sea capaz de realizar las siguientes cosas desde un menú de opciones:

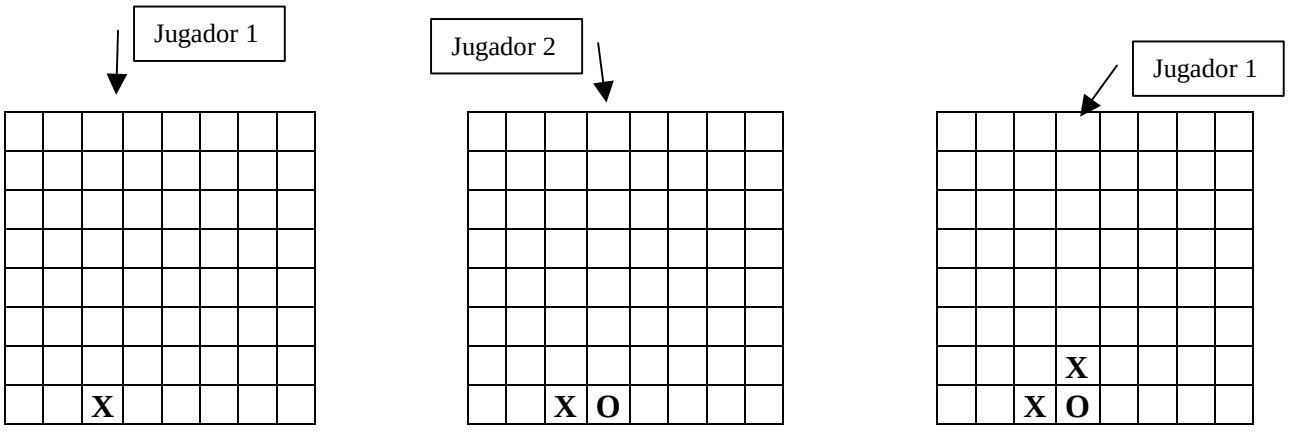
1.- mostrar el vector actual
2.- mostrar su valor máximo y en qué posición se encontraba
3.- la suma de todos sus elementos
4.- la media aritmética de todos sus elementos
5.- modificar un elemento dado del vector
0.- salir

2.- Hacer un programa que permita realizar las siguientes operaciones con matrices de números enteros, desde el siguiente menú de opciones.

1.- introducir 2 matrices A y B. (Se deberá usar 2 veces la misma función. Esta misma función o procedimiento también se encargará de pedir número de filas y columnas de las matrices).
2.- calcular la suma A+B (si es posible) y mostrar el resultado
3.- calcular la resta A-B (si es posible) y mostrar el resultado
4.- calcular el producto A*B (si es posible) y mostrar el resultado
0.- salir

3.- Generar una tabla de 101 filas con los valores de x, sin x, cos x, tan x, donde x variará desde 0 a PI. Es decir para los siguiente valores de x: 0, PI/100, 2PI/100, ...99PI/100, PI). El programa deberá mostrar en cada pantalla 10 filas de datos y preguntar si sigue presentando la siguiente pantalla o no.

4.- Programa el juego del 4 en raya. Se trata de un juego de 2 jugadores (A y B) con 2 tipos de fichas distintas que se colocan sobre un tablero 8 x 8. El jugador que tiene el turno elige en qué columna quiere poner su ficha y esta cae desde la fila 8 hasta la fila 1 del tablero o hasta que encuentre la primera fila no ocupada. El juego termina cuando un jugador es capaz de poner 4 fichas en raya (hecho que debe ser detectado por el programa) o cuando se llena toda el tablero o simplemente cuando se desee acabar la partida.



Duración de la práctica: 1 sesión