# Fundamentos de Programación

## Relación de Problemas
## Tema 4

**1.- Diseñe la clase matrices que viene a continuación:**

```
#define M 10
class matrices {
        int tabla [M];
public:
        void cargar();
                //Pondrá en cada elemento de la tabla el valor de su índice.
        void encontrar();
                //Pedirá un numero entero por teclado y mostrara por pantalla si ese
                //numero esta en la tabla o no
};
```

**2.- Diseñe la clase matrices que viene a continuación:**

```
#define M 10
#define N 15
class matrices {
        int tabla [M] [N];
public:
        void cargar();
                //Pondrá en cada elemento de la tabla el valor de la suma de sus índices.
        void encontrar();
                //Pedirá un numero entero por teclado y mostrara por pantalla si ese
                //numero esta en la tabla o no
};
```

**3.- Diseñe la clase matrices que viene a continuación:**

```
#define M 10
#define N 15
typedef char cadena [30];

class matrices {
        cadena tabla [M] [N];
public:
        void cargar();
                //Pondrá en cada elemento de la tabla una palabra leída desde teclado.
        void encontrar();
                //Pedirá una palabra por teclado y mostrara por pantalla si esa palabra esta
                //o no en la tabla y en que fila y columna se encuentra
};
```

**4.- Diseñe la clase matrices que viene a continuación:**

```
#define M 10
#define N 15
typedef char cadena [30];
struct persona {
        long dni;
        cadena nombre;
        };

class matrices {
        persona tabla [M] [N];
public:
        void cargar();
        //Pondrá en cada elemento de la tabla un dni y un nombre leídos desde teclado.
        void encontrar();
                //Pedirá un dni por teclado y mostrara por pantalla si ese dni esta o no en la
                //tabla y además en que fila y columna se encuentra
};
```

**5.- Realice una clase que invierta el contenido de un vector conocido.**

```
#define N 15
class vector {
        int uno [N];
        int num;
public:
        void cargar ( );
                //Pedirá por teclado números enteros poniéndolos en la tabla uno.
                //num será la cantidad de números almacenados.
        void invertir ();
                //Pondrá el primer elemento de uno como ultimo, el segundo como
                //penúltimo y así…
};
```

**6.- Determinar si dos tablas tab1 y tab2 tienen el mismo contenido. Diseñe una clase que resuelva este problema.**

```
class vector {
        int tab1[10], tab2[10];
public:
        void cargar ();
                //Llenara las dos tablas con valores leídos desde teclado.
        int comparar ();
                //devolverá un 1 si son diferentes tab1 y tab2, 0 si son iguales
 };
```

**7.- Determinar si una matriz es simétrica. Diseñe una clase que resuelva este problema.**

```
class matriz {
        int   a[7][7];
public:
        void cargar ();
                //Llenara la tabla a completa con valores leídos desde teclado.
        bool simétrica ();
                //devolverá true si la matriz a NO es simétrica, false en caso contrario
};
```

**8.- Realice una clase de tal forma que fusione dos tablas (ordenadas previamente) en otra tercera, de tal forma que los elementos queden ordenados en ésta.**

```
class vector {
        int  uno [15], dos[15], fus[30];
        int numuno, numdos;
public:
        void cargar ();
                //Cargará la tabla uno con  valores leídos desde teclado, el número de
                //elementos leídos se pondrá en numuno. Cargará la tabla dos con valores
                //leidos desde teclado, el número de elementos leídos se pondrá en numdos,
                //se supone que los valores se ponen ordenados crecientemente desde teclado.
        void mezclar ();
                //Cargará la tabla fus con los valores de uno y dos quedando la tabla fus
                //ordenada crecientemente.
};
```