

### Enunciado:

Resuelva los siguientes ejercicios sobre contenedores de la librería estándar de C++: `std::map` y `std::set`. Utilice el estándar C++14 en la solución de sus problemas. No olvide compilar con los *flags* apropiados para detectar *warnings* y errores.

### Map

1. Para la clase `map` explore el uso de métodos para:
  - a) Agregar parejas (llave,valor) al mapa
  - b) Eliminar parejas por llave
  - c) Acceder al valor asociado a una llave
  - d) Usar `operator[]`
  - e) Determinar el número de elementos en el mapa
2. Escriba un programa que
  - a) Cree un objeto de la clase `map<string,string>`
  - b) Agregue al mapa elementos donde la llave corresponda a una palabra y el valor a su definición
  - c) Use `operator[]` para acceder al valor asociado a algunas llaves
  - d) Elimine un elemento del mapa usando su llave
  - e) Recorra e imprima todos los elementos del mapa (utilice un iterador, primero con un ciclo `for` y luego con un `while`)
3. [*Phone book.*] Use `map` para crear un directorio telefónico de profesores del Departamento MACC de la Universidad. Implemente dos funciones que hacen uso de un `map` tal que se permite buscar ya sea por número de la extensión o por las iniciales del profesor. Su solución debe satisfacer los siguiente requerimientos:
  - a) La primera función debe tomar como parámetro las iniciales del profesor y retornar el número de la extensión. Si el profesor, es decir las iniciales no se encuentran dentro del directorio, la función debe retornar `-1`.
  - b) La segunda función debe tomar como parámetro un número de extensión y retornar las iniciales del profesor. Si la extensión no se encuentra dentro del directorio, la función debe retornar el `string` vacío `""`.
  - c) En la generación del directorio telefónico, use el siguiente texto

Ext	Prof
4052	ACD
4012	CEAC
4010	EJAL
4044	GO

4053	J J J R
4013	J F P B
4009	P A M G
4037	R A M R
4008	V G U

Cree un archivo de texto, que contiene esta información, y léalo desde su programa para crear el correspondiente directorio que debe usar dentro de las funciones mencionadas anteriormente.

Considere las iniciales como variables de tipo `string` y los números de extensiones como variables de tipo `short`. Demuestre el funcionamiento de su implementación invocando las dos funciones desde la función principal.

4. El código Morse es un método para transmitir información escrita en forma de tonos o clics que pueden ser largos (rayas) o cortos (puntos). Escriba un programa que lee del teclado un mensaje del usuario y lo traduce a o descifra de código Morse. Su programa debe tener las siguientes características:
  - a) Escriba el código que codifica letras a puntos y rayas en un archivo de texto que debe ser leído. Esta información debe ser guardada en un `map`. ¿Qué tipos de datos deben ser la llave (*key*) y el valor (*value*)?
  - b) Si la línea comienza con una letra, el mensaje debe ser traducido a código Morse. Cualquier carácter que no corresponda con las letras del alfabeto simplemente debe ser ignorado.
  - c) Si la línea comienza con un punto o una raya, debe leerse como una serie de caracteres del código Morse que debe ser traducido a letras.
  - d) Cada secuencia de puntos y rayas está separada por espacios, pero cualquier otro carácter debe ignorarse.
  - e) Debido a que no hay codificación para el espacio entre palabras, los caracteres del mensaje traducido se imprimirán sin dejar espacio.
  - f) El programa debe finalizar la lectura de un mensaje cuando el usuario ingrese una nueva línea `<enter>`. Sin embargo, el programa siempre debe estar listo para recibir un nuevo mensaje.
  - g) Use el símbolo `'>'` para mostrarle al usuario que está listo para recibir un nuevo mensaje.
  - h) No olvide manejar apropiadamente las letras mayúsculas y minúsculas.
  - i) Una vez que su programa está funcionando correctamente agregue el código para los números.

Por ejemplo, su programa debería ejecutar de la siguiente manera:

```
Traductor de/a clave Morse
> SOS TITANIC
... --- ... - .. - .- -. .. -.-
```

```
> .... --- .-. .- .--- .-. .- -. ...--  
HOLA JUAN3  
>
```

## Set

5. Para la clase `set` explore el uso de métodos para:
  - a) Agregar elementos al conjunto
  - b) Eliminar elementos del conjunto
  - c) Determinar si el conjunto está vacío
  - d) Buscar un elemento por su valor
6. Escriba un programa que
  - a) Cree un objeto de la clase `set<string>`
  - b) Agregue al conjunto 5 nombres diferentes
  - c) Elimine un elemento del conjunto usando su valor
  - d) Busque un elemento que está en el conjunto
  - e) Busque un elemento que NO está en el conjunto
  - f) Imprima el número de elementos y cada uno de los elementos en el conjunto
  - g) Obtenga e imprima el primer y último elementos en el conjunto
7. Dados dos contenedores asociativos (*associative containers*) tipo `std::set`, implemente una función predicado `subset(set<int> & a, set<int> & b)` que toma como parámetros dos referencias a conjuntos de enteros `a`, `b`, y retorna `true` si el `a` es un subconjunto de `b`. En caso contrario el predicado debe retornar `false`.
8. Suponga que dos conjuntos pueden ser representados computacionalmente como `vector`s de enteros. Utilizando `set` para registrar el resultado de la correspondiente operación, y solamente los métodos exportados por esta librería, implemente la unión, la intersección y la diferencia de conjuntos. Su prototipo debe ser

```
1 | set<int> aunion(vector<int> & a, vector<int> & b);  
2 | set<int> intsect(vector<int> & a, vector<int> & b);  
3 | set<int> difference(vector<int> & a, vector<int> & b);
```

¿Cuáles otras operaciones sobre conjuntos se pueden implementar usando `set`?