

Tarea 16

Las preguntas precedidas por un asterísco son para los alumnos de maestría. En licenciatura, dan puntos extras.

1 Patrones

Ejercicio 1 *¿Por qué no compila el código siguiente? En particular, explicar las diferencias entre las dos últimas líneas de la función main().*

```
class Toto {};  
template<class T> class Truc {  
    int i;  
public:  
    template <class U>  
    void someMethod(Truc<U>& s) {  
        s.i = 3;  
    }  
};  
int main() {  
    Truc<int> a;  
    Truc<Toto> b, c;  
    a.someMethod(c);  
    b.someMethod(c);  
}
```

Ejercicio 2 *Encontrar en la documentación de la librería estándar C++ el patrón map, explicar a partir de la documentación para que sirve y cuales son los parametros del patrón y para que sirven.*

***Ejercicio 3** *Re-implementar la clase patrón Stack de la clase usando un list < T > al lugar de un arreglo para almacenar los datos. Consultar la documentación de list en la librería estándar para aprender a usar este contenedor.*

Ejercicio 4 *Escribir una función patrón con un solo parámetro tipo T. La función tomará como argumentos (1) un arreglo de T, (2) un índice (entero) de inicio, (3) un índice (entero) de fin, y (4) un valor inicial (que por default estará igual al objeto por default resultando del constructor por default de T, T()). La función calculará la suma de todos los T del arreglo entre los dos índices especificados, sumada con el valor inicial pasado en cuarto argumento. Escribir un ejemplo con generaciones de las clases para T=int, double, string.*

Ejercicio 5 *Escribir una clase patrón para manejar matrices, e.g. el patrón se podrá llamar `Matrix`. Las clases de este patrón permitirán (1) acceder a los elementos de la matriz, (2) sumar y multiplicar una matriz a la matriz llamando el método y (3) multiplicar por un escalar la matriz llamando el método.*