

Tarea 19

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

1 Funciones-algoritmos, objetos funciones

***Ejercicio 1** *Escribir una función patrón ISORTED() que tomará como entradas dos iteradores (como los ejemplos que vimos en la clase) y que regresará true si los datos del contenedor correspondiente son ordenados (en orden creciente o decreciente), false sino. Escribir un programa ejemplo del uso de ese patrón.*

Ejercicio 2 *Re-escribir el código siguiente usando las funciones-algoritmos de la clase, de tal manera que desaparezcan los ciclos y que se pueda redefinir los dos parametros del modulo (aquí, el 7 y el 3) en la ejecución misma.*

```
int tab[]={12,33,-7,6,18,-37};
int s=sizeof(tab)/sizeof(tab[0]);
vector<string> l;
for (int i=0;i<s;i++) {
    if (tab[i]%7<3)
        l.push_back("NO");
    else
        l.push_back("SI");
}
for (list<string>::iterator lit=l.begin();lit!=l.end();lit++)
    cout << *lit << endl;
```

Ejercicio 3 *A partir de un ejemplo y de la documentación de la STL verificar si la función REMOVE() aplicada con nada más un par de iteradores,*

```
template <class It, class T>
It remove(It first, It last, const T& value);
```

*cambia o no el tamaño del contenedor cuyos iteradores son pasados de argumentos. Intentar dar una explicación de por qué es así, y deducir que entonces la función REMOVE() o la función UNIQUE() con estos tipos de argumentos sólo pueden estar aplicadas a contenedores **ordenados**. Explicar finalmente como funciona UNIQUE().*

Ejercicio 4 *Escribir un programa que recupere el nombre de un archivo como parámetro de línea de comando. Abrir este archivo, leerlo palabra por palabra (operador >>). Guardar esas palabras en un vector de string (que eventualmente tendrás que personalizar para manejar lo de minúsculas/mayúsculas). Luego:*

1. *ordenar todas esas palabras (sin tomar en cuenta lo minúsculo/mayúsculo) con la función-algoritmo SORT(),*
2. *quitar las occurencias múltiples con la función UNIQUE(),*
3. *quitar las occurencias de menos de tres letras,*
4. *imprimir el resultado.*

Ejercicio 5 *Escribir una funcion transform_if() que actúe un poco como la primera forma de TRANSFORM() de la clase (es decir una función unaria sobre elementos del primer contenedor) pero (1) con una función binaria sobre dos objetos (habrá que pasar entonces **dos** pares de iteradores para los objetos input) y (2) de tal manera que sólo los elementos satisfaciendo un predicado están transformados hacia elementos del segundo contenedor. Se regresará un iterador end() del contenedor de resultado (que podrá ser de tamaño mas chico que los dos de entrada).*