ESTRUCTURA DE DATOS

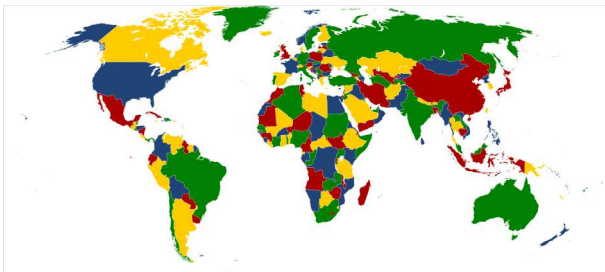
Ejercicio 1 *(1.2) Construir una plantilla que permita almacenar una matriz de datos. Completar la siguiente declaración:*

```
template <class T>
class Matriz:private vector<vector<T> >
{
private:
    int n_filas;
    int n_cols;
public:
 //constructor
 //sobrecargar >>, <<
 //retorna el valor de la fila i y la columna j
 T celda(int i, int j);
 //retorna un vector con la diagonal principal(si es cuadrada)
 vector<T> diagonal_principal();
//retorna la matriz formada por todas las columnas desde col_inicial hasta col_final
//(col_inicial<=col_final)
 Matriz<T> sub_matriz(int col_inicial, int col_final);
 };
```

Leer una matriz desde un archivo (usando >>) y verificar el funcionamiento de la clase construida.

Ejercicio 2 *(1) Considere la siguiente conjetura:*

Dado cualquier mapa geográfico con regiones continuas, entonces el mapa puede ser coloreado con cuatro colores diferentes, de forma que no queden regiones adyacentes con el mismo color. Asumiremos que las regiones adyacentes comparten no solo un punto, sino todo un segmento de borde (frontera) en común.
En la figura se puede observar un mapa coloreado con 4 colores. Se desea escribir un programa que permita almacenar una instancia del problema anterior.



1. *Un país puede ser almacenado como un pair $< string, int >$, donde la primera componente almacena el nombre y la segunda la población.*

2. *Completar la siguiente clase:*

```
class paises:private list<pair<string,int> >
{
....
};
```

3. *Leer un conjunto de países desde un archivo usando >>. El archivo tiene el formato:*
   nombre_pais población
   ....

4. *Escribir una función miembro que permita ordenar los países crecientemente respecto a su población.*

5. *Escribir una función miembro que determine el rango de la población de los países.*

6. *Escribir una función miembro que reciba un entero $2 \leq k \leq 10$ y divida los países almacenados en k grupos de acuerdo a su población respecto al siguiente criterio:*

   Grupo 1: $min \leq poblacion < \frac{rango}{k}$

   Grupo 2: $\frac{rango}{k} \leq poblacion < 2\frac{rango}{k}$

   ...

   Grupo k: $(k-1)\frac{rango}{k} \leq poblacion$

   Si cada grupo es una cola, entonces los grupos anteriores pueden ser almacenados en la estructura:

$$vector < queue < pair < string, int >>>$$