



3. Vectores

Importar el proyecto de la clase. En CLion seleccionar File → New CMake Project From Sources... y seleccionar la carpeta con el nombre Labo03-Vectores que se encuentra en el archivo Labo03-Vectores.zip. En onlinegdb copiar la estructura de los archivos main.cpp, ejercicios.cpp y ejercicios.h.

Ejercicio 1. Implementar los siguientes problemas:

1. `bool divide(vector<int> v, int n)` Dados un vector v y un entero n , decide si n divide a todos los números de v .
2. `int maximo(vector<int> v)` que dada una secuencia de enteros, devuelve el valor máximo.
3. `bool pertenece(int elem, vector<int> v)` que dado un entero, indica si pertenece o no a la secuencia.
4. `void mostrarVector(vector<int> v)` que dada una secuencia de enteros, la muestra por la salida estándar (cout). Ejemplo: si la secuencia es $\langle 1, 2, 5, 65 \rangle$ se debe mostrar en pantalla $[1, 2, 5, 65]$
5. `void sinImpares(vector<int>& v)` que dada una secuencia de enteros, devuelve el mismo vector colocando un 0 en las posiciones en las que haya un número impar.
6. `vector<int> limpiarDuplicados(vector<int> v)` que dada una secuencia de enteros, devuelve un vector de enteros con los elementos de la secuencia sin duplicados. Ejemplo $v = \langle 1, 1, 2, 1, 1, 2, 3, 2, 3, 3 \rangle$ el resultado es $\langle 1, 2, 3 \rangle$
7. `vector<int> rotar(vector<int> v, int k)` que dada una secuencia v y un entero k , rotar k posiciones los elementos de v . $[1, 2, 3, 4, 5, 6]$ rotado 2, debería dar $[3, 4, 5, 6, 1, 2]$.
8. `vector<int> reverso(vector<int> v)` que dada una secuencia v , devuelve el reverso. Implementar también la versión recursiva de este problema.
9. `vector<int> factoresPrimos(int n)` que dado un entero devuelve un vector con los factores primos del mismo. Los factores primos de un número entero son los números primos divisores exactos de ese número entero. Ejemplos: los factores primos de 6 son 2 y 3. Factores primos de 7 es 7
10. `bool estaOrdenado(vector<int> v)` que dada una secuencia v de int, dice si es monótonamente creciente o monótonamente decreciente.
11. `void negar(vector<bool>& v)` Modifica un vector de booleanos negando todos sus elementos.
12. `vector<pair<int, int> > cantidadApariciones(vector<int> v)` que dada una secuencia de enteros, devuelve una tupla que contine por cada entero la cantidad de apariciones del mismo. Ejemplo $v = \langle 1, 1, 2, 1, 1, 2, 3, 2, 3, 3 \rangle$ el resultado es $\langle (1, 4), (2, 3), (3, 3) \rangle$

Tip: para aprender a usar las tuplas entrar a <http://www.cplusplus.com/> y buscar (en donde dice search) pair.

Ejercicio 2. *Vectores y entrada salida.* Implementar las siguientes funciones. Se puede usar como inputs los archivos en la carpeta Labo03-Vectores/archivos

1. `void palindromos(string rutaArchivoIn, string rutaArchivoOut)` Este procedimiento debe leer un archivo que contiene una lista de strings y crear uno nuevo dejando sólo los palíndromos. Además, debe transformar las palabras a mayúscula. **Ayuda:** Buscar la función `toupper` definida en `cctype`. Utilizar como ejemplo el archivo `palindromos.txt`.
2. `void promedios(string rutaArchivoIn1, string rutaArchivoIn2, string rutaArchivoOut)` Dados dos archivos en los que cada uno contiene una secuencia de reales (ambas con la misma longitud), guardar el promedio de cada par de números que se encuentran en la misma posición en el archivo de salida. Por ejemplo: si tenemos dos secuencias "1.5 2 3 4" y "1.5 25 3 12" el resultado debe ser "1.5 13.5 3 8"
3. `void cantidadApariciones(string rutaArchivoIn, string rutaArchivoOut)` Dado un archivo `rutaArchivoIn`, que contiene una lista de números separados por espacios, contar la cantidad de apariciones de cada uno y escribe `rutaArchivoOut` con una línea por cada número encontrado, un espacio y la cantidad de apariciones. Por ejemplo: si el "1" aparece 44 veces y el "2" 20 veces, la salida debería contener dos líneas: "1 44" y "2 20".

4. `void estadisticas(string rutaArchivo)` Dado un archivo de texto, mostrar por pantalla las estadísticas de cantidad de palabras con longitud 1,2,3... hasta el máximo. Por ejemplo:

```
Palabras de longitud 1: 100
Palabras de longitud 2: 12
Palabras de longitud 3: 6
...
```

5. `void interseccion()` Procedimiento que pide al usuario que se ingresen dos nombres de archivos que contengan sólo números enteros separados por espacios, luego calcula la intersección (números que se encuentran en ambos archivos) e imprime por pantalla el resultado.