- String, Array y Vector -

NOTA: Debe entregar el código completo de la librería (edya_stl.h). Recuerden seguir los criterios aconsejados por la cátedra y respetar los prototipos de las funciones. **Colocar en el encabezado de todos los archivos el nombre del Grupo** (X es el nombre del grupo).

```
// GRUPO=X
```

Ejercicio 1

Utilizando el tipo de dato abstracto **string** provisto por C++ se pide listar todas las letras del string utilizando:

1. El método at().
2. El operador de indexación.
3. Iteradores.
4. Bucle en un rango (Range-based for loop, C++11).

Ejercicio 2

Utilizando el tipo de dato abstracto **string** provisto por C++ se pide implementar una función que genere a partir de un string palíndromo un nuevo string con todos los substring palíndromos concatenados en forma de pirámide.

```
string palindromePyramid(const string& str);
```

Para la resolución debe utilizar los constructores y métodos de la clase string por ejemplo `append`, `length`, `substr`, etc. Es importante que lea la documentación de string. Por ejemplo,

```
str = palindromePyramid("reconocer");
// str == "    n    \n   ono   \n  conoc  \n econoce \nreconocer"
```

La salida debe ser igual al ejemplo, si en una línea se agregan por ejemplo 3 espacios al principio también se deben agregar al final.
Salida por consola:

```
    n
   ono
  conoc
 econoce
reconocer
```

Ejercicio 3

Utilizando el tipo de dato abstracto **string** provisto por C++ se pide implementar una función que genere un nuevo string reemplazando todas las ocurrencias del substring `from` por el substring `to` respetando el siguiente prototipo:

```
string replaceAll(const string& str, const string& from, const string& to);
```

Por ejemplo:

1. `replaceAll("reconocer", "r", "R") == "ReconoceR"`
2. `replaceAll(replaceAll("reconocer", "r", "R"), "e", "E") == "REconocER"`

Ejercicio 4

Utilizando el tipo de dato abstracto `vector` provisto por C++ se pide:

1. Implementar una función que genere en forma aleatoria un vector de `N` números enteros positivos cuyo valor oscila en rango indicado `[from, to]`.

```
vector<unsigned int> create(unsigned int N, unsigned int from, unsigned int to);
```

2. Codifique una función que imprima los elementos de un vector de enteros en dos formatos. El primer formato es uno por cada línea (`FMT_LINE`) de pantalla usando `endl` y el segundo en notación de lista (`FMT_LIST`) [elemento1, elemento2, …., elementon], esté es el valor por defecto.

```
void print(vector<unsigned int> v, format f=FMT_LIST)
```

3. Implementar una función que ordene ascendentemente el vector de enteros pasado por parámetro.

```
void sort(vector<unsigned int> &v);
```

No se puede usar sort de STL.

4. Implementar una función que permita crear un nuevo vector insertando los elementos de un vector dentro de otro en la posición indicada.

```
vector<unsigned int> insert(vector<unsigned int> v1, vector<unsigned int> v2,
unsigned int pos);
// Ejemplo si: v1=[1, 2, 3, 4], v2=[6, 7, 8]
v3 = insert(v1, v2, 1)
// v3 = xxx[1, 6, 7, 8, 2, 3, 4]

// Ejemplo si: v1=[1, 2, 3, 4], v2=[6, 7, 8]
v3 = insert(v1, v2, 111)
// v3 = xxx[1, 2, 3, 4, 6, 7, 8]
```

5. Implementar una función que liste los números contenidos en el vector y la cantidad de ocurrencias.

```
void print_frequency(vector<unsigned int> v);
```

Ejercicio 5

A partir de la union `rgb` implementar una matriz por medio de la clase `array` que pasada por parámetro a la función `dibujar(...)` genere un archivo .ppm de 100px X 100px con alguno de los siguientes colores Deep Purple, Indigo, Amber, Blue Gray y Light Green y una cruz sobre él haciendo contraste según las guías de diseño de Material Design.

```
void dibujar(array<array<rgb,TAM>,TAM> image, string filename){
    const int dimx = image.size();
    const int dimy = image.size();
    ofstream outfile (filename, ofstream::binary);
    outfile << "P6\n" << dimx << " "<< dimy<< "\n255\n";
    for(array<rgb,TAM> f: image)
        for(rgb c: f)
            outfile << char(c.r) << char(c.g) << char(c.b);
    outfile.close();
}
```

Nota: se utiliza el formato binario P6.

Ejemplo **Light Green**

Ejemplo **Blue Grey:**

.

Los editores de imagen normalmente reconocen el formato PPM, en caso de no tener ninguno compatible pueden usar GIMP.