

Proyecto 2

Recursión

Objetivo

Definir funciones recursivas.

Descripción

Se requiere escribir 4 funciones que resuelvan problemas en forma recursiva. Las 4 funciones son:

- Dado un string, regresar el string “limpio”, lo que significa que si hay caracteres adyacentes iguales, sólo deja uno de ellos.
 - Por ejemplo, “aabbbbccccdd”, debe regresar “abcd”.
- Contar el número de veces que un substring dado aparece en un string.
 - e.g. “ab” en el string “abbaaabba”, debe regresar un 2.
- Dado un entero no negativo, regresar la suma de sus dígitos.
 - e.g. “12345” debe regresar 15.
- Dado un string formado por paréntesis anidados, regresar TRUE si están anidados correctamente o FALSE en caso contrario.
 - e.g. “(())” regresa TRUE, “((()”) regresa FALSE.

API

Aunque las funciones no se relacionan una con otra, sólo para efectos de entrega del proyecto, coloque las 4 funciones en una clase llamada **Recursion**.

El proyecto se puede hacer con cada función solicitada directamente recursiva o con cada función como interface para llamar a una función recursiva que reciba los parámetros que le hagan falta. En cualquier caso, las funciones del API deben ser públicas y las otras privadas.

La API del proyecto debe quedar de la siguiente forma, considerando los cambios de acuerdo al lenguaje utilizado (e.g. **string** para C++ y **String** para Java):

```
// recibe un string y regresa el string “limpio”  
string limpiaString(string s);
```

```
// recibe un string s y un substring sub y regresa el número de veces que  
// aparece el substring en el string.  
int cuentaSubstring(string s, string sub);
```

```
// recibe un entero n y regresa la suma de sus dígitos
int sumaDigitos(int n);

// recibe un string s formado con paréntesis anidados y regresa true si los
// paréntesis están correctamente anidados y false si no lo están.
bool anidacionCorrecta(string s);
```

Requerimientos

- La definición de la clase se debe hacer EXACTAMENTE con los **nombres y tipos** que se especifican en el API solicitada.
- El nombre del archivo donde se guardará la clase debe ser **Recursion.h** o **Recursion.java**. Este archivo deberá contener toda la definición de la clase y los **#include** o **import** necesarios para que funcione el proyecto.
- Las pruebas se harán por medio de un archivo **Main.cpp** o una clase **Main.java** con el siguiente formato:

C++	Java
<pre>#include "Recursion.h" int main() { ... }</pre>	<pre>public class Main { public static void main(String args[]) { ... } }</pre>

Por lo que el archivo entregado **no deberá contener un main** si está en C++.

- El cuerpo de la función o método **main** del archivo de prueba contendrá **definiciones de objetos** de la clase pedida y algunas **operaciones** sobre dichos objetos para probar los métodos solicitados en el API.
- Para hacer las pruebas de funcionamiento, tanto el archivo **Recursion.h** o **Recursion.java**, como el de prueba se colocarán en el mismo folder.

Entrega

- Se deberán entregar dos archivo:
 - Uno **.h** o **.java**, que contendrá la definición de la clase (**Recursion.h** o **Recursion.java**).
 - Uno **.docx** o **.pdf**, que contendrá la documentación con TODAS las partes solicitadas y descritas en el syllabus.
- Ambos archivos se subirán a la liga correspondiente al “Proyecto 2”, que se encuentra en Bb en la sección “Proyectos”.
- **SÚBALOS POR SEPARADO EN EL MISMO SUBMIT. NO HAGA UN ZIP DE LOS DOS ARCHIVOS.**