

HERRAMIENTAS COMPUTACIONALES

SEMANA 6. FUNCIONES Y RECURSIVIDAD

SECCIÓN-05

2018-2

La solución debe subirse a SicuaPlus en un único archivo `.py` con el nombre `NombreApellido_hw3.py` el cual debe contener toda la solución del taller.

Un programador desea verificar la sintaxis de su programa en términos de corchetes `{}` antes de compilarlo. Para esto se debe comprobar que por cada corchete abierto existe un corchete cerrado y que no deben quedar corchetes sueltos. Por ejemplo `{{}}{}}` tiene sintaxis correcta, mientras que `{}}{}}` tiene sintaxis incorrecta.

La idea es eliminar recursivamente corchetes cerrados `}` hasta que no se pueda más. En caso de que todos los corchetes sean eliminados, la sintaxis es definida como correcta. En caso contrario, la sintaxis es definida como incorrecta.

1. (1.5 points) Digna la variable `codigo` de tipo string, la cual contiene corchetes `{,}` y otro tipo de strings. Por ejemplo:

```
codigo = "for(int i = 0; i<100; i++){if(lista[i]<=3){print(i);} else{print(0);}}"
```

Luego, defina una función llamada `filtro(inp)` que tome como parámetro un string `inp` y retorne el string que queda al eliminar todos los caracteres que o sean corchetes. Recuerde que los strings no tienen métodos modificadores asociados y es útil definir otra variable donde se guarde el proceso de eliminación. Por ejemplo, el resultado al aplicar está función en `codigo` es:

```
filtro(codigo)->"{{}}{}}"
```

2. (1.5 points) Escriba una función `borrar(inp)` que tome como parámetro un string y retorne el string que queda al eliminar el **PRIMER** par de corchetes cerrados `}` del string. Si no hay un par de corchetes cerrados, la función debe retornar la cadena `inp` intacta. Por ejemplo:

```
borrar("{{}}{}}")->"{{}}{"  
borrar("{}{")->"{}{"
```

Debe utilizar la función `filtro` antes de aplicar `borrar`.

Quizas le sea de ayuda la función `str.replace()`.

3. (2.0 points) Ahora defina la función `recursive_borrar(inp)` que tome como parámetro un string y elimine recursivamente todos los corchetes cerrados `}` hasta que no queden más por eliminar. Esta función debe mostrar el resultado de la eliminación recursiva de corchetes cerrados. Por ejemplo:

```
recursive_borrar("{{}}{}}")->""  
recursive_borrar("{{}}{}}{}")->"{}{"
```

Teniendo en cuenta que sí el resultado de esta función recursiva es vacío, la sintaxis de la variable entrada es correcta, imprima un mensaje en pantalla que diga:

El programa revisado tiene sintaxis (in)correcta