

# Proyecto 1

## Definiciones Recursivas

### Descripción

Hacer un programa que reciba una definición recursiva de un lenguaje **L**, un número **N**, y genere todos los strings que se generen aplicando **N** veces el paso recursivo.

### Entrada

La definición recursiva de un lenguaje se debe recibir en dos partes:

- 1) Todos los casos base
- 2) Todos los casos del paso recursivo

El programador deberá decidir cómo recibir cada uno de esos datos (e.g. por medio de un archivo, solicitando al usuario que escriba de determinada forma, etc).

Para este proyecto vamos a simplificar las expresiones que se pueden dar en el paso recursivo:

- Solamente pueden tener operaciones de concatenación. Por ejemplo, dados  $u, v \in \Sigma^*$ , las expresiones posibles serán **uv, uav, auv, uva, aubvc**, etc., donde  $a, b, c \in \Sigma$ .
  - En realidad, también está permitiendo operaciones de unión las cuales se tienen con las opciones del paso recursivo.
- Las variables que representen strings que pertenecen al lenguaje sólo pueden ser **{u,v,w,x,y,z}**.
- Los símbolos del alfabeto sólo podrán ser de un solo carácter.

### Salida

El programa debe entregar como salida todos los strings que se generan aplicando el paso recursivo **N** veces.

En cada paso sólo puede usar los strings generados en cualquiera de los pasos anteriores, no se usan los generados en el mismo nivel.

### Comentarios

- Es individual
- Puede usar su lenguaje de programación favorito
- Debe cumplir con la documentación (que ahora es muy simple).

En la siguiente hoja vienen 2 ejemplos.

### Ejemplos

Ejemplo 1	Ejemplo 2
<p>Si la entrada es: Paso Base: <math>\lambda</math> Paso Recursivo: <math>u0, 1u</math> <math>N = 3</math></p> <p>La salida sería: Aplicando el paso recursivo: 0 veces: <math>\lambda</math> 1 vez: 0, 1 2 veces: 00, 10, 11 3 veces: 000, 110, 100, 111</p>	<p>Si la entrada es: Paso Base: <math>\lambda</math> Paso Recursivo: <math>uu0, w1u</math> <math>N = 2</math></p> <p>La salida sería: Aplicando el paso recursivo: 0 veces: <math>\lambda</math> 1 vez: 0, 1 2 veces: 000, 110, 010, 111, 011, 01, 11, 10</p>