

# Ejercicios de teoría del Tema 2

## *Compiladores*

Dpto. de Ingeniería de la Información y las Comunicaciones



## Ejercicio 1

### **Enunciado:**

Definir una expresión regular para reconocer todos los identificadores formados por letras minúsculas excepto "if".

## Ejercicio 1

### Enunciado:

Definir una expresión regular para reconocer todos los identificadores formados por letras minúsculas excepto "if".

### Solución:

- $L : a|...|z$
- $LsinI : a|...|h|j|...|z$
- $LsinF : a|...|e|g|...|z$
- **ident** :  $LsinI.L^* \mid i.(LsinF.L^*|\lambda) \mid i.f.L^+$

## Ejercicio 2

### **Enunciado:**

Un identificador es una secuencia de letras, dígitos y caracteres guión bajo que comienza por una letra. No se permiten dos caracteres guión bajo seguidos y tampoco pueden tener un guión bajo al final.

## Ejercicio 2

### Enunciado:

Un identificador es una secuencia de letras, dígitos y caracteres guión bajo que comienza por una letra. No se permiten dos caracteres guión bajo seguidos y tampoco pueden tener un guión bajo al final.

### Solución:

- $L : a|...|z|A|...|Z$
- $D : 0|...|9$
- $LD : L \mid D$
- **ident** :  $L.LD^*.(\_LD^+)^*$

## Ejercicio 3

### Enunciado:

Divide el siguiente programa C++ en los tokens adecuados.

¿Qué tokens tendrían valores léxicos (lexemas) asociados?

¿Cuáles serían dichos lexemas?

---

```
float limitedSquare(float x) {  
    /* returns x-squared, but never more than 100 */  
    return (x <= -10.0 || x >= 10.0) ? 100 : x*x;  
}
```

---

## Ejercicio 3

### Solución:

---

```
[float,] [id,1] [(,] [float,] [id,2] [],] [{,] [return,] [(,]
[id,2] [leq,] [minus,] [litfloat,3] [or,] [id,2] [geq,]
[litfloat,3] [],] [?,] [litint,4] [:,] [id,2] [mult,] [id,2]
[;,] [},]
```

---

Tabla de símbolos:

1	limitedSquare
2	x
3	10.0
4	100