

TEMA 3: PROGRAMACIÓN EN LENGUAJES DE ALTO NIVEL

CUESTIONES

1. Indicar cuáles de los siguientes identificadores no son válidos en C++.

- a) while1
- b) while
- c) X143
- d) 2Sumas
- e) Media_Parcial
- f) Media_Total
- g) sin
- h) Suma*N
- i) Año

2. ¿Cuáles de las siguientes definiciones de tipo no son válidas?

- a) enum Ing {Sal, Aceite, Lechuga};
- b) typedef UnaLetra char;
- c) enum Vocal {'A', 'E', 'I', 'O', 'U'};
- d) typedef Natural unsigned int;

3. Dadas las siguientes declaraciones:

```
float R; int I; char C;
```

Indicar cuáles de las siguientes expresiones dan error de compilación:

- a) $I = \text{int}(C) + I;$
- b) $R = I / 10;$
- c) $R + I = \text{int}(C);$
- d) $I = R \% 100;$
- e) $R = \cos(R) / \ln(\text{fabs}(R));$
- f) $C = \text{char}(I * 3);$

4. En las siguientes sentencias para hallar la media de dos números reales existe algún error. ¿De qué tipo es?

```
x = 10; y = 0;
media = x - y / 2;
```

- a) Error sintáctico.
- b) Error de ejecución.
- c) Error lógico.
- d) Hay dos errores: uno sintáctico y otro lógico.

5. ¿Qué le sucede a la variable lógica "Booleana" tras esta sentencia ?

```
Booleana = 5.0 > 1
```

- a) Nada. Da un error de no coincidencia de tipos.
- b) Queda como *true*.
- c) Queda como *false*.
- d) Nada. Da un error de ejecución.

6. ¿Qué diferencia hay en principio para el compilador entre estos dos grupos de sentencias?

```
{ cout << "Valor:"; { cout << "Valor:";
  cout << 0.0; ;      cout << 0.0;
}                    }
```

- a) el grupo de la izquierda tiene una sentencia más pero es vacía.
- b) el grupo de la izquierda da un error del tipo: "sentencia esperada".
- c) a) y b) son ciertas.
- d) a) y b) son falsas.

7. Evaluar paso a paso y decir el tipo resultante de las siguientes expresiones:

- a) $4 + 5 / 2$
- b) $8 * 3 / 5 \% 3$
- c) $10 - \text{float}(7) / 2$
- d) $\text{int}(5 + 3.0)$
- e) $(0 < 0 / 2) \|\| (1 == 5 \% 2) \&\& ! (2 + 2 * 3 < 10)$
- f) $(5 + 2 \% 2 == 0) \|\| (1 <= 3 / 2) \&\& (\text{int}(7.5) > 7)$

8. Dada la siguiente gramática:

```
<frase> ::= (<dato>)
<datos> ::= $<cifra> | $<cifra><dato>
<cifra> ::= 0|1|2|3|4|5|6|7|8|9
```

Indica cuales de las siguientes frases son incorrectas (no pertenecen a la gramática):

- a) (\$1)\$4
- b) (\$1+\$0)
- c) (\$2\$3)
- d) (\$101)