

EXAMEN PRIMERA SEMANA DE FEBRERO (TIPO A)

Pregunta 4.

4. La sentencia del lenguaje C±:

```
scanf("mes%2d", &mes);
```

- a) Se ejecuta correctamente si se lee el dato: mes3
- b) Siempre se ejecuta correctamente
- c) Se ejecuta correctamente si se lee el dato: 03
- d) No se ejecuta correctamente si se lee el dato: mes1234

Comentarios:

Pregunta sobre el mecanismo de lectura disponible en C±. El funcionamiento de scanf se introduce en la página 65 del libro de texto de Fundamentos de Programación. En el punto 4 se dice: “Un carácter no blanco en la cadena de formatos hace que se lea (y se salte) el siguiente carácter de la entrada, que debe coincidir exactamente con el carácter del formato”.

Además, en las páginas 382 y 383 del libro, se puede ver un claro ejemplo del funcionamiento de scanf con formato. En este ejemplo, se leen puntos con el formato (x,y), usando para ello la sentencia “scanf (“(%g, %g)”, &x, &y);”, indicándose que se produce una excepción cuando en la lectura de un punto aparece un dato que no es una representación válida de un punto (x,y), con los paréntesis y la coma de separación. Por tanto, se deja claro que para que scanf funcione correctamente, la cadena de entrada que lee, tiene que coincidir con el formato que tiene asociado.

En la sentencia mostrada en la pregunta 4 se está indicando a scanf que lea datos de entrada que coincidan con el formato “mes” seguido de una serie de dígitos, pero de manera que se almacenen en la variable *mes* como máximo 2 de ellos. Así:

- a) *CORRECTA. Se ejecuta correctamente si se lee el dato: mes3.* Esta respuesta es correcta, ya que el dato de entrada coincide con el formato de scanf.
- b) *ERRONEA. Siempre se ejecuta correctamente.* Esta respuesta es errónea, ya que si el dato de entrada no comienza por “mes”, no se ejecutará correctamente.
- c) *ERRONEA. Se ejecuta correctamente si se lee el dato: 03.* Esta respuesta es errónea, ya que el dato de entrada no comienza por “mes”
- d) *ERRONEA. No se ejecuta correctamente si se lee el dato: mes1234.* Esta respuesta es errónea, ya que el dato de entrada coincide con el formato de scanf, esto es, es “mes” seguido de una serie de números enteros y, por tanto, scanf se ejecuta correctamente, almacenando en la variable *mes* el valor 12, ya que le habíamos indicado que almacenara como mucho dos dígitos.