

PEQUEÑAS CUESTIONES DE PROGRAMACIÓN

Estos ejercicios corresponden al tema 4, en el que no se dispone de mas estructura compuesta de Pascal que la secuencia, y se ha hablado de tipos de datos entero, real y lógico, con sus operaciones, así como de la sentencias de lectura y escritura.

1. ¿Qué sucede, en la implantación de Pascal que se utilice, cuando se usa una variable sin tener valor asignado?
2. ¿Cuál es el valor de MAXINT? (en la implantación de Pascal que se utiliza).
3. Escribir el siguiente programa. Antes de compilarlo, calcular el valor de $3*MAXINT$.

```
program Q3 (output);
var n : integer;
Begin
    n := 3*MAXINT;
    writeln (n);
End.
```

Compilar el programa. Leer los mensajes de compilación. ¿Se ha producido un ejecutable? Si es así, ejecutarlo y comparar con el valor calculado.

Borrar el ejecutable y compilar con la opción `-Cr`. Observar ahora los mensajes de compilación y comprobar que no se ha generado ejecutable.

4. Escribese el siguiente programa:

```
program Q4 (output);
var n : integer;
Begin
    writeln (2147483647, ' ', 2147483647+5);
    write ('MAXINT y MAXINT*MAXINT: ':25);
    writeln (maxint , ' ', maxint*MAXINT);

    n := MAXINT;
    write ('n := MAXINT; n y n*n: ':25);
    writeln (n, ' ', n*n);

    n := n*n;
    write ('n:= n*n; valor de n:':25);
    writeln (n);
End.
```

Calcular el resultado manualmente. Compilar después sin opciones (`fpc ComoSeLlame.pas`). ¿Qué se obtiene en la ejecución?

Compilar después con las opciones `-Cr -gl`. Comprobar las diferencias. ¿Cuál es la explicación?

5. ¿Qué resultado se obtiene para `x DIV y` en los siguientes casos?

a) $y < 0 \wedge x > 0$

b) $y > 0 \wedge x < 0$

c) $y < 0 \wedge x < 0$

6. ¿Cuál es el valor de las expresiones siguientes?

a) `trunc (1.6)`

e) `round (1.6)`

b) `trunc(-1.6)`

f) `round(-1.6)`

c) `trunc(-1.4)`

g) `round(-1.4)`

d) `trunc(-1.5)`

h) `round(-1.5)`

7. Especificar la postcondición de la sentencia compuesta

```
begin
  a := b;
  b := a
end
```

si la precondition es $\{a = a_0 \wedge b = b_0\}$

8. Especificar la postcondición de la sentencia compuesta

```
begin
  a := a + b ;
  b := a - b ;
  a := a - b
end
```

si la precondition es $\{a = a_0 \wedge b = b_0\}$

9. Obténganse expresiones en Pascal para calcular:

a) el dígito de las unidades de un entero positivo n

b) el dígito de las centenas de un entero positivo n

c) la suma de un entero n y tantas decenas como indique el dígito de sus centenas

d) la parte entera de un número positivo x

e) la parte fraccionaria de un número positivo x

f) el número entero más cercano al número positivo x

g) el valor redondeado a tres decimales del número positivo x

h) el valor en euros de una cantidad entera de pesetas n (con el redondeo correspondiente)

10. Dado un número entero positivo de exactamente tres cifras (entrada), escríbase un programa Pascal que escriba en pantalla sus cifras en orden inverso.

11. Escríbase un programa Pascal que, dada como entrada una cantidad positiva de euros (con céntimos), muestre el número mínimo de billetes y monedas posible en que podría descomponerse.

12. Escribese un programa en Pascal que, tomando como entrada los valores de dos ángulos, expresados en grados, minutos y segundos, obtenga el valor de su suma de la misma forma. Se supone que las entradas son correctas, en el sentido de que cada ángulo está expresado como tres números enteros, en los rangos respectivos $[0, 360)$, $[0, 60)$ y $[0, 60)$. La salida debe estar expresada de la misma manera.
13. Escribese un programa Pascal que, tomando como entrada un número de 3 cifras (multiplicando) y otro de 2 (multiplicador) muestre el producto de la forma en que se vería si se hiciera a mano. Por ejemplo, dados 739 y 12, debe mostrar en pantalla:

```

      739
    x  12
    -----
      1478
      739
    -----
     8868

```

¿Podría generalizarse para un multiplicando positivo cualquiera? ¿Y para un multiplicador positivo cualquiera?

14. ¿Es cierto que $a*b \text{ DIV } c = a*(b \text{ DIV } c)$?
15. Expresar en Pascal las expresiones lógicas siguientes (interpretése menor y mayor como menor estricto y mayor estricto respectivamente, positivo como mayor o igual que 0 y negativo como menor o igual que 0):
 - a) x es menor que 5
 - b) x es mayor que 5 y menor que 7
 - c) x es menor que 5 o mayor que 7
 - d) x es mayor que 5 y no mayor que 7
 - e) x es mayor que 5 y no menor que 7
 - f) x es menor que 5 y no menor que 7
 - g) ni x ni y son nulos
 - h) al menos uno de los dos es nulo (con referencia a x e y)
 - i) x e y tienen el mismo signo (considérese que el 0 tiene signo positivo y negativo)
 - j) x e y tienen el mismo signo y ninguno es nulo
 - k) entre x e y uno es negativo pero no ambos
 - l) al menos uno de los dos es negativo
 - m) a lo sumo uno de los dos es negativo
 - n) entre x e y , al menos uno vale 7 y su producto no llega a 122
 - ñ) al menos uno de los dos es positivo y al menos uno de los dos es par
 - o) x , y y z tienen el mismo valor

- p*) x, y y z tienen valores distintos
- q*) x, y y z están en orden creciente
- r*) c (de tipo `char`) es una letra minúscula

16. Especifíquese, para las siguientes expresiones, el tipo que deben tener todas las variables que intervienen para que la expresión sea correcta, y el valor, cuando sea posible:

- | | |
|--|--|
| <i>a</i>) $x = y$ | <i>f</i>) $p = q$ or r |
| <i>b</i>) $\text{odd}(k)$ or $\text{odd}(\text{succ}(k))$ | <i>g</i>) $\text{odd}(n*(n-1))$ |
| <i>c</i>) $p = \text{true}$ | <i>h</i>) $\text{ord}('b') - \text{ord}('a') > 0$ |
| <i>d</i>) $10 \text{ div } 3 = 10/3$ | <i>i</i>) $\text{sqr}(a) < 0$ |
| <i>e</i>) $p > \text{succ}(p)$ | <i>j</i>) $1 = 1$ |

17. Hállense expresiones para calcular:

- a*) el número de vueltas de un ángulo dado en grados
- b*) el cuadrante (1, 2, 3 ó 4) que alcanza un ángulo dado en grados
- c*) el equivalente en radianes de un ángulo dado en grados
- d*) el equivalente en grados de un ángulo dado en radianes
- e*) la tangente de un ángulo dado en grados
- f*) el valor numérico de un dígito almacenado como carácter
- g*) el carácter en mayúsculas de uno dado en minúsculas

18. Escribir el programa que se especifica a continuación:

Entrada : Tres números reales

Salida : Si se puede formar un triángulo con esos valores, el mensaje
"Se puede formar triángulo= TRUE"

Si no, el mensaje

"Se puede formar triángulo= FALSE"

Si se puede formar, el mismo tipo de salida para "Equilátero", "Isósceles" y "Escaleno",
("Equilátero = TRUE" ...)

Suposiciones: Las entradas son todas estrictamente positivas

Ejemplos: para 5 3.0 4 en la entrada, la salida será:

Se puede formar triángulo = TRUE

Equilátero = FALSE

Isósceles = FALSE

Escaleno = TRUE

Para 5 2 2 en la entrada, la salida será:

Se puede formar triángulo = FALSE