

## Ejercicios sobre punteros

1. Dadas las siguiente declaraciones:

```

TYPE Indice = 0..9;
ApuntIndice = ^Indice;
VAR
    I: Indice;
    ApuntI: ApuntIndice;

```

- a) ¿Qué contiene entonces `ApuntI`?  
b) Si en seguida ejecutamos la codificación:

```

new(ApuntI);
ApuntI^:=2;
I := 4;

```

¿qué contendrá entonces `ApuntI`? ¿Qué contendrá `ApuntI^`?

2. Después de ejecutarse el fragmento de codificación

```

TYPE
    Cosa = integer;
    ApuntadorACosa = ^Cosa;
VAR
    C, CC : Cosa;
    ApuntC, ApuntCC : ApuntadorACosa;

BEGIN
    ApuntC := NIL;
    new(ApuntCC);

```

¿cuáles de las siguientes variables contienen basura?

- a) `ApuntC`  
b) `ApuntCC`  
c) `C`  
d) `CC`  
e) `ApuntC^`  
f) `ApuntCC^`
3. Prepara un diagrama de apuntes que muestre la situación después de ejecutarse el fragmento de código del ejercicio anterior.

4. Si después de ejecutar la codificación del ejercicio 2, el programa ejecuta también el código `ApuntCC:=3;`, ¿cuál será entonces la situación? Dibuja un diagrama de apuntadores.
5. Supongamos que

```
VAR
  Eso: ^Char;
```

¿Será entonces legal llamar a `New(Eso^)`? ¿Y llamar a `New(Eso)`? Explícalo.

6. Supongamos que

```
TYPE
  Acertijo = real;
  ApAcert = ^Acertijo;
VAR
  A1, A2 : ApAcert;
```

¿cuáles de los siguientes enunciados serán legales?

- a) `A1:=1.1;`
- b) `A1:=1.1^;`
- c) `new(A1);`
- d) `A1:= NIL;`
- e) `A1^:=1.1;`
- f) `new(A1^);`
- g) `A2:=A1;`
- h) `A2:=^1.1;`
- i) `A2:=^A1;`

7. ¿Qué exhibe el siguiente programa?

```
PROGRAM QuePasa;
TYPE
  ApuntadorC = ^Char;
VAR
  A1, A2: ApuntadorC;
BEGIN
  new(A1);
  new(A2);
  A1^:= 'A';
  A2^:= 'B';
  writeln(A1^);
  writeln(A2^);
END.
```

8. Dadas las siguientes definiciones y declaraciones

```

type
  TPEntero = ^integer;
  TPCharacter = ^char;
var
  P1, P2 : TPEntero;
  Q1, Q2 : TPcharacter ;

```

¿cuál es la salida de los siguientes fragmentos?

- a)           new(P1);  
               new(P2);  
               P1^:= 6;  
               P2^:= P1^+20;  
               writeln('P1^ igual a ', P1^, ' P2^ igual a ', P2^);
- b)           new(P2);  
               P2^:= 2;  
               P2^:= sqr(P2^);  
               new (P1);  
               P1^:= P2^ mod 3;  
               writeln('P1^ igual a ', P1^, ' P2^ igual a ', P2^);
- c)           new(Q1);  
               new(Q2);  
               Q1^:='Z';  
               Q2^:= pred(Q1^);  
               writeln('Q1^ igual a ', Q1^, ' Q2^ igual a ', Q2^);

9. Dadas las mismas declaraciones del ejercicio anterior, describir y corregir los errores sintácticos de los siguientes fragmentos:

- a)           new(P1);  
               new(P2);  
               new(Q1);  
               readln(Q1^);  
               P2^:= P1;  
               writeln('Q1^ igual a ', Q1^, ' Q2^ igual a ', Q2^);
- b)           new(P1);  
               P1:= P2;  
               P1:= 3.5 \* P1^;
- c)           new(P1);  
               new(Q2);  
               P1^:= 48;  
               Q2^:= char(P1^);  
               P1:= Q2;

10. Dadas las siguientes declaraciones:

```
var
    x: integer;
    P1, P2: ^integer;
    Q1, Q1: ^real;
```

¿Qué es incorrecto (si lo hay) en cada una de las sentencias?

- a) `writeln(P1);`<sup>1</sup>
- b) `P1:= Q1;`
- c) `if P1^ = nil then Q1:= Q2;`
- d) `readln(P1^)`
- e) `new(X)`
- f) `begin`  
     `P1^:= 17;`  
     `new(P1);`  
   `end`

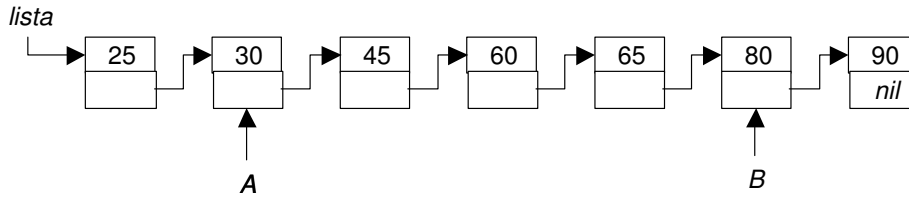
11. ¿Cuál es la salida del siguiente trozo de código?

```
program punteros;
type
    estudiante = record
        Letra:char;
        Edad: integer;
    end;
    punt_estu = ^estudiante;
var
    p1, p2: punt_estu;
begin
    new(p1);
    p1^.edad:=10;
    p1^.letra:='A';
    new(p2);
    p2^.edad:=11;
    p2^.letra:='C';
    writeln (p1^.letra, p1^.edad, p2^.letra, p2^.edad);
    p1^.edad:= p2^.edad;
    p1^.letra:= p2^.letra;
    writeln (p1^.letra, p1^.edad, p2^.letra, p2^.edad);
    p1:=p2; {ojo, esto es lícito, pero deja a p1^sin liberar}
    writeln (p1^.letra, p1^.edad, p2^.letra, p2^.edad);
end.
```

---

<sup>1</sup>Algunos compiladores podrían dar error. No el FreePascal. Sí daría un error de ejecución `readln (P)`

12. Dada la siguiente lista:



donde

```

type
  tinfo= integer;
  tenlace= ^tnodo;
  tnodo= record
    info: tinfo;
    sig: tenlace;
  end;
  tcoleccion: tenlace;
var
  lista: tcoleccion;
  A, B: tenlace;

```

a) Dar los valores de las siguientes expresiones:

- 1)  $A^{\wedge}.info$
- 2)  $B^{\wedge}.sig^{\wedge}.info$
- 3)  $Lista^{\wedge}.sig^{\wedge}.sig^{\wedge}.info$

b) Verdadero o falso:

- 1)  $Lista^{\wedge}.sig = A$
- 2)  $A^{\wedge}.sig^{\wedge}.info = 60$
- 3)  $B^{\wedge}.sig = nil$

c) Indicar si la sintaxis de las siguientes sentencias son correctas o no, y explicar cuál es el problema, si lo hay.

- 1)  $Lista^{\wedge}.sig := A^{\wedge}.sig$
- 2)  $Lista^{\wedge}.sig := B$
- 3)  $Lista^{\wedge}.info:=B$
- 4)  $B:=A^{\wedge}.sig^{\wedge}.info$
- 5)  $Lista:=B^{\wedge}.sig^{\wedge}.sig$
- 6)  $B:=B^{\wedge}.sig^{\wedge}.sig^{\wedge}.sig$

d) Escribir una sentencia para cada una de las siguientes acciones:

- 1) Hacer que Lista apunte al nodo que contiene 45
- 2) Hacer que B apunte al último nodo de la lista

3) Hacer que Lista apunte a una lista vacía

e) Mostrar lo que escribe el siguiente segmento de código

```
Var Ptr: tenlace;
    lista: tcoleccion;
...
New(lista);
New(Ptr);
Lista^.info:=2;
Ptr^.info:=5;
Lista^.sig:=Ptr;
Lista:=Ptr;
writeln(Ptr^.info, Lista^.info);
```

f) Mostrar lo que escribe el siguiente segmento de código

```
Var Ptr: tenlace;
    lista: tcoleccion;
...
New(lista);
Lista^.info:=10;
New(Ptr);
Ptr^.info:=18;
Ptr^.sig:=nil;
Lista^.sig:=Ptr;
New(Ptr);
Ptr^.info:=20;
Ptr^.sig:=lista;
Lista:=Ptr;
while Ptr<>NIL do begin
    writeln(Ptr^.info);
    Ptr:=Ptr^.sig;
end; (*while*)
```