



FACULTADE DE INFORMÁTICA
Departamento de Computación
Programación II

Ejercicios de punteros

1. Suponga que tenemos las declaraciones:

```
type
  tIndice = 0..9;
  tPIndice = ^tIndice;
var
  i : tIndice;
  apuntI : tPIndice;
```

- 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. Sea el siguiente fragmento de codificación:

```
1      type
2          tPEntero = ^integer;
3      var
4          c, cc: integer;
5          apuntC, apuntCC: tPEntero;
6      begin
7          apuntC:= nil;
8          new(apuntCC);
```

Contestar a las siguientes preguntas:

- a) Después de la ejecución de la línea 8, ¿cuáles de las siguientes variables contienen basura?
- I) `apuntC`
 - II) `apuntCC`
 - III) `c`
 - IV) `cc`
 - V) `apuntC^`
 - VI) `apuntCC^`
- b) Representar gráficamente la situación de todas las variables-enteras, punteros y dinámicas.
- c) Si añadimos la sentencia `apuntCC^:= 3`; ¿cuál será entonces la situación? Dibuje un diagrama de apuntadores.

3. Suponga que:

```

type
    tReal = real;
    tPReal = ^tReal;
var
    A1, A2: tPReal;

```

¿Cuáles de los siguientes enunciados serán válidos en ese caso?

- 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;

4. ¿Qué exhibe el siguiente programa?

```

program QuePasa;
type
    tPCharacter = ^char;
var
    A1, A2: tPCharacter;
begin
    new(A1);
    new(A2);
    A1^:='A';
    A2^:='B';
    A1:= A2;
    writeln(A1^);
    writeln(A2^);
end;

```

5. Dadas las siguientes definiciones y declaraciones:

```

type
    tPEntero = ^integer;
    tPCharacter = ^char;
var
    P1, P2 : tPEntero;
    Q1, Q2 : tPCharacter;

```

¿qué hacen los siguientes fragmentos de código?

- 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^:= chr(P1^);
P1:= Q2;
```

d)

```
new(P1);
new(Q2);
P1^:= 6;
Q2^:= chr(P1^+59);
writeln('P1^ igual a ', P1^, ' Q2^ igual a ', Q2^);
```

e)

```
new(Q1);
new(Q2);
Q1^:='Z';
Q2^:= pred(Q1^);
writeln('Q1^ igual a ', Q1^, ' Q2^ igual a ', Q2^);
```

6. Asumir las siguientes declaraciones:

```
type
  tPInteger = ^integer;
  tPReal = ^real;
var
  X: integer;
  P1, P2: tPInteger;
  Q1, Q2: tPReal;
```

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

a) `writeln(P1);`

b) `P1:= Q1;`

c) `if P1^ = nil then Q1:= Q2;`

d) `readln(P1^)`

e) `new(X)`

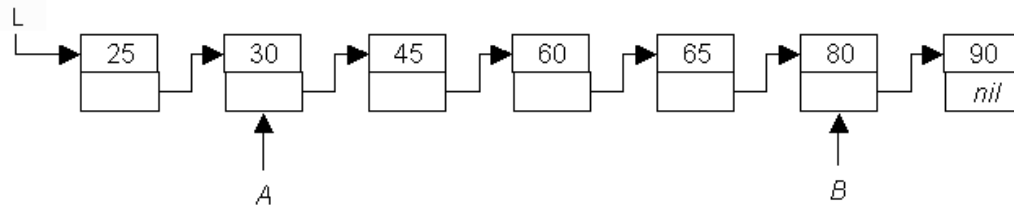
f)

```
P1^:= 17;
new(P1);
```

g)

```
new(P1);
P1^:= 17;
new(P1);
```

7. Sea la siguiente figura que representa una estructura enlazada que almacena una colección de elementos:



y sean:

```

type
  tInfo = integer;
  tEnlace = ^tNodo;
  tNodo = record
    info: tInfo;
    sig: tEnlace;
  end;
  tColeccion = tEnlace;
var
  L: tColeccion;
  A, B: tEnlace;

```

a) Dar los valores de las siguientes expresiones:

- I) $A^{\wedge}.info$
- II) $B^{\wedge}.sig^{\wedge}.info$
- III) $L^{\wedge}.sig^{\wedge}.sig^{\wedge}.info$

b) Decir si se verifican las siguientes igualdades:

- I) $L^{\wedge}.sig = A$
- II) $A^{\wedge}.sig^{\wedge}.info = 60$
- III) $B^{\wedge}.sig = nil$

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

- I) $L^{\wedge}.sig := A^{\wedge}.sig$
- II) $L^{\wedge}.sig := B$
- III) $L^{\wedge}.info := B$
- IV) $B := A^{\wedge}.sig^{\wedge}.info$
- V) $L := B^{\wedge}.sig^{\wedge}.sig$
- VI) $B := B^{\wedge}.sig^{\wedge}.sig^{\wedge}.sig$ (nil no tiene siguiente)

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

- I) Hacer que L apunte al nodo que contiene 45
- II) Hacer que B apunte al último nodo de la lista
- III) Hacer que L apunte a una colección vacía

e) Mostrar lo que escribe el siguiente segmento de código:

```

var
  Ptr : tEnlace;
...
new(L);
new(Ptr);
L^{\wedge}.info:= 2;
Ptr^{\wedge}.info:= 5;
L:=Ptr;
Ptr^{\wedge}.info:= 7;
writeln(Ptr^{\wedge}.info, L^{\wedge}.info);

```

f) Mostrar lo que escribe el siguiente segmento de código:

```
var
    Ptr : tEnlace;
...
new(L);
L^.info:= 10;
new(Ptr);
Ptr^.info:= 18;
Ptr^.sig:= nil;
L^.sig:= Ptr;
New(Ptr);
Ptr^.info:= 20;
Ptr^.sig:= L;
L:= Ptr;
while Ptr <> nil do begin
    writeln(Ptr^.info);
    Ptr := Ptr^.sig;
end; (*while*)
```

8. Dadas las declaraciones siguientes:

```
type
    tPNumero = ^tNodeNumero;
    tNodeNumero = record
        datos: integer;
        sig: tPNumero;
    end;
var
    P1, P2: tPNumero;
    P3: ^integer;
```

y suponiendo que se han ejecutado previamente las instrucciones

```
new(p1); new(p2); new(p3);
```

¿Qué resultado se visualizará en este fragmento (en caso de error, indicar cuál)?:

```
p1^.datos:= 12;
p2^.datos:= 34;
p1^.sig:= p2;
writeln(p2^.datos);
writeln (p2^.sig^.datos);
```

9. ¿Cuál es la salida correcta del siguiente programa?:

```
1      program ejemplo_punteros;
2
3      var
4          p, q, r : ^integer;
5
6      begin
7          new(p);
8          p^:= 10;
9          q:= p;
10         q^:= q^ + 1;
11         dispose(q);
12         q:= nil;
13         { ATENCION: p ya no apunta a una direccion valida }
```

```

14
15         new(q);
16         { ATENCION: ahora q apunta una NUEVA direccion valida,
17           pero p sigue apuntado a una direccion no valida}
18
19         q^:= 20;
20         writeln('valor de q^ = ', q^);
21         writeln('valor de p^ = ', p^);
22     end.

```

- a) Se produce un error de ejecución en la línea 21.
- b) Se imprime: valor de $p^{\wedge} = 10$
- c) Se imprime: valor de $p^{\wedge} = 11$
- d) Se imprime: valor de $p^{\wedge} = 20$