UNIVERSIDADE DA CORUÑA

FACULTADE DE INFORMÁTICA
Departamento de Computación
Estructura de Datos e da Información

Ejercicios de punteros

1. Suponga que tenemos las 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);
ApntI^:= 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. Preparar un diagrama de apuntadores que muestre la situación de inicialización después de ejecutarse el fragmento de codificación del ejercicio 2

4. Si después de ejecutar la codificación del ejercicio 2, el programa ejecuta la sentencia `ApuntCC^ := 3;` ¿cuál será entonces la situación de inicio? Dibuje un diagrama de apuntadores.

5. Suponga que

```
var
    Eso: ^char;
```

¿Será legal entonces llamar a `New(Eso^)`? ¿Y llamar a `New(Eso)`? Explique por qué.

6. Suponga que

```
type
    Acertijo = real;
    ApAcert = ^Acertijo;
var
    a1, a2: ApAcert;
```

¿Cuáles de los siguientes enunciados serán legales 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;`

7. ¿Qué exhibe el siguiente programa?

```
Program QuePasa;
type
    ApuntadorC = ^Char;
var
    A1, A2: ApuntadorC;
begin
    New(A1);
    New(A2);
    A1^:='A';
    A2^:='B';
    A1:= A2;
    writeln(A1^);
    writeln(A2^);
end;
```

8. Dadas las siguientes definiciones y declaraciones

```
type
    TPEntero = ^integer;
    TPCaracter = ^char;
var
    P1, P2 : TPEntero;
    Q1, Q2 : TPcaracter ;
```

¿cuál es la salida de 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^:= char(P1^);
P1:= Q2;
```

9. Asumir las siguientes declaraciones:

```
var
    x: integer;
    P1, P2: ^integer;
    Q1, Q2: ^real;
```

¿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)`

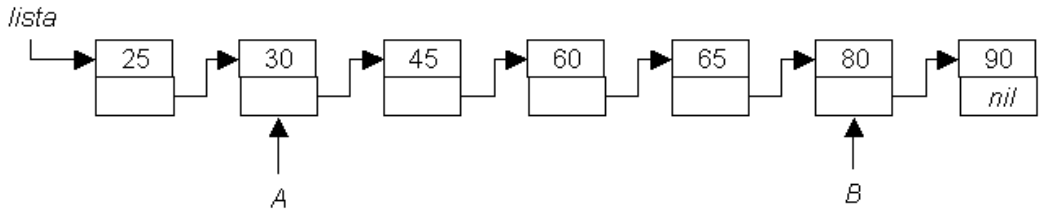
f)

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

10. ¿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.
```

11. Sea la siguiente figura que representa una lista:



y sean:

```
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^.info`
2) `B^.sig^.info`
3) `Lista^.sig^.sig^.info`

b) Decir si se verifican las siguientes igualdades:

1) `Lista^.sig = A`
2) `A^.sig^.info = 60`
3) `B^.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^.sig : = A^.sig`
2) `Lista^.sig := B`
3) `Lista^.info := B`
4) `B := A^.sig^.info`
5) `Lista:=B^.sig^.sig`
6) `B := B^.sig^.sig^.sig` (nil no tiene siguiente)

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;
...
New(lista);
New(Ptr);
Lista^.info:=2;
Ptr^.info:=5;
Lista:=Ptr;
Ptr^.info:=7;
writeln(Ptr^.info, Lista^.info);
```

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

```
var
    Ptr : tEnlace;
...
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*)
```

12. Dadas las declaraciones siguientes:

```
type
    PteroNumero = ^NodoNumero;
    NodoNumero = record
                    Datos: integer;
                    Sig: PteroNumero;
                 end;
var
    P1, P2: PteroNumero;
    P3: ^integer;
```

y suponiendo que se han ejecutado previamente las instrucciones

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

¿Qué resultado se visualizará en cada uno de los siguientes fragmentos (en caso de error, indicar cuál):

a)

```
p1^.Datos := 123;
p2^.Datos := 456;
p1^.sig := P2;
writeln(p1^.Datos);
writeln (p1^.sig^.Datos);
```

```
p1^.Datos := 12;
p2^.Datos := 34;
p1:=p2;
writeln(p1^.Datos);
writeln (P2^.Datos);
```

b)

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

d)

```
p1^.Datos := 12;
p2^.Datos := 34;
p3^.Datos := 34;
p1^.sig := p2;
p2^.sig := p3;
writeln(p1^.Datos);
writeln(p2^.Datos);
writeln(p3^.Datos);
```