



HOJA DE PROBLEMAS
PUNTEROS

P.1. Deducir la salida del siguiente código:

```
Program puntero;  
Type  
    Pprint: ^integer;  
Var  
    I,J:Pprint;  
    N:integer;  
Begin  
    New(I);  
    New(j);  
    N:=5;  
    I^:=N;  
    Writeln(I^);  
    J:=I;  
    J^:=-7;  
    Writeln(I^)  
End.
```

P.2. ¿Cuál es la salida del siguiente programa?

```
Program test;  
Type  
    Estudiante=record  
        Letra:char;  
        Edad:integer  
    End;  
    Puntest: ^Estudiante;  
Var  
    P1,P2:Puntest;  
Begin  
    New(P1);  
    P1^.Edad:=1;  
    P1^.Letra:= 'A'  
    Writeln(P1^.Edad,P1^.letra);  
    New(P2);  
    P2^.Edad:=1;  
    P2^.Letra:= 'A'  
    Writeln(P2^.Edad,P2^.letra);
```

```

P1:=P2;
P2^.Edad:=3;
P2^.Letra:='C';
Writeln(P1^.Edad, P1^.Letra,P2^.Edad,P2^.Letra)

```

End.

P.3. Deducir la salida del siguiente código:

```

Program pruebasimple;
Var
    I,J:^integer
    x,y:integer;
Begin
    New(I);
    New(j);
    I^:=5;
    x:=I^+1;
    y:=I^;
    New(J)
    J^:=y+3
    Writeln(x, ' ', y, ' ', I^, ' ', J^)
End.

```

P.4. Si p y q son ambos de tipo cadena (^string) ¿Qué se imprimirá al ejecutar este segmento de programa?.

```

New(p);
New(q);
p^:='Carpe';
q^='Diem';
q:=p;
q^:='concha';
Writeln(p^);
Writeln(q^);

```

P.5. Escribir una función que devuelva como valor el producto de dos polinomios así como un programa que la utilice. Los polinomios serán representados mediante arrays de tal forma que el grado cero del polinomio se almacene en la posición 1 del vector y así sucesivamente. La estructura de datos será:

```

type
tdimp=0..10;
tdimpt=0..20;
tArray=Array[tdimpt] of real;
tApArray=^tArray;

```

Las funciones y procedimientos para implementar el programa principal tendrán las siguientes cabeceras:

```

procedure inicia_polinomios(var a,b:tAparray; var dima,dimb:tdimp);

```

```

{Efecto. pide los grados de los polinomios e inicializa los vectores a cero}
procedure pedir_datos(var a,b:tApArray; dima,dimb:tdimp);
{Efecto. Pide los coeficientes de los polinomios}
function prod_pol(a,b:tApArray;dima,dimb:tdimp):tApArray;
{Dev. Los coeficientes del polinomio producto}
Procedure visualiza(c:tApArray;dimt:tdimpt);
{Efecto. Visualiza los coeficientes del polinomio en cuestión}

```

P.6. Supóngase que se dispone de la siguiente declaración de tipos y constantes:

```

TYPE
    Tnombre=string[20];
    Thobbies=(futbol,bailar,viajar,caminar);
    TConjHobbies=SET OF Thobbies;
    TPunPersona=^Tpersona;
    Tpersona=Record
        Nombre:Tnombre;
        Hobbies:^TconjHobbies;
    End;

```

Escribir los procedimientos para almacenar los datos asociados a una persona P:TPunPersona; y destruir los datos asociados a una persona P:TpunPersona teniendo cuidado de no dejar sin liberar toda la memoria ocupada. Las cabeceras de los procedimientos serán:

```

PROCEDURE CrearDatosPersona (VAR P:TpunPersona);

PROCEDURE BorrarDatosPersona (VAR P:TPunPersona);

```

Nota: Utilizar un procedimiento que ya está implementado denominado LeerConjHobbies(VAR s:TConjHobbies) tal que carga un conjunto de hobbies en "s".

P.7. Supóngase que se dispone de la siguiente declaración de tipos y constantes:

```

CONST
    TopeTabla=20;
TYPE
    Tnombre=string[20];
    TpunNombre=^Tnombre;
    Tindice=1..TopeTabla;
    Ttabla=array[TIndice] of TpunNombre;

```

Suponiendo que la tabla está llena, escribir un procedimiento en Pascal que detecte cuál es el nombre que debería ocupar la primera posición de la tabla (según el orden alfabético) y lo intercambie con el ubicado en la primera posición, si es que se da el caso. Se supone que todos los punteros de la tabla apuntan a un nombre. La cabecera del procedimiento será:

```

PROCEDURE ubicamenornombre ( VAR tabla:Ttabla)

```