

PRACTICA N° 8
PUNTEROS Y TIPOS DE DATOS RECURSIVOS

1) Dadas las siguientes declaraciones

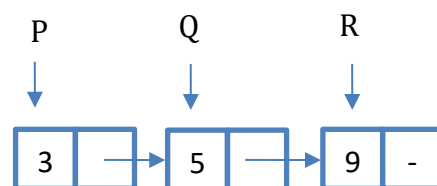
```
TYPE
    TPUNTERO = ^INTEGER;
VAR
    P, Q : TPUNTERO;
```

- a- Escriba un procedimiento en Pascal para intercambiar los valores de los apuntadores P y Q, de tal modo que después que el procedimiento se efectúe, P apuntará al elemento al cual Q apuntaba antes y viceversa.
 - b- Escriba un procedimiento en Pascal que haga que P apunte al mismo elemento al que Q apunta, y se deshaga del elemento al que P apuntaba anteriormente.
 - c- Escriba un procedimiento en Pascal que produzca una nueva variable, con P apuntando hacia ella y con el mismo contenido del elemento al que Q apunta.
- 2) Mostrar lo que harían, gráficamente, las siguientes sentencias en el diagrama esquemático que se adjunta.

```
TYPE
    TPUNTERO = ^NODO ;
    NODO = RECORD
        INFO : INTEGER ;
        SIG :TPUNTERO
    END;
VAR
    P , Q , R : TPUNTERO ;
```

...

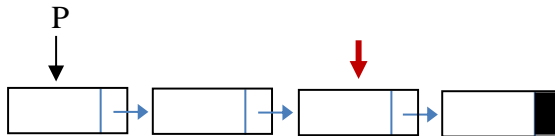
```
P := P^.SIG
Q := P
R := P^.SIG
P^.INFO := Q^.INFO
P^.INFO := Q^.SIG^.INFO
```



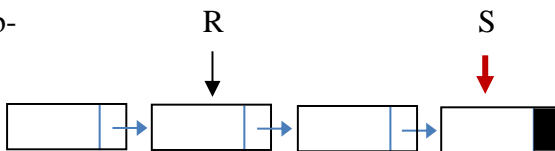
$R^{\wedge}.SIG := P$

- 3) Escribir UNA SOLA sentencia, usando la notación \wedge , para efectuar el cambio indicado por la línea gruesa (roja). P debe apuntar finalmente al mismo espacio que la línea gruesa.

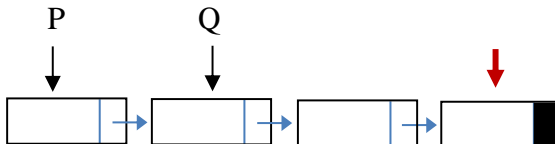
a-



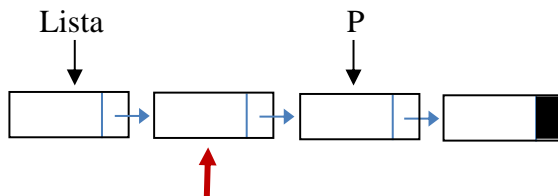
b-



c-



d-



4) Mostrar que escriben en la salida estándar los siguientes segmentos de código:

a) `NEW (P);`
 `NEW (Q);`
 `P^.INFO := 5 ;`
 `Q^.INFO := 6 ;`
 `P := Q ;`
 `P^.INFO := 1 ;`
 `WRITELN (P^.INFO , Q^.INFO);`

b) `NEW(P);`
 `P^.INFO := 3;`
 `NEW(Q);`
 `Q^.INFO := 2;`
 `NEW(P);`
 `P^.INFO := Q^.INFO;`
 `Q^.INFO := 0;`
 `WRITELN(P^.INFO , Q^.INFO);`

c) `NEW(P);`
 `NEW(Q);`
 `P^.INFO := 0;`
 `P^.SIG := Q;`
 `Q^.SIG := NIL;`
 `Q^.INFO := 5;`
 `P^.SIG^.SIG := P;`
 `Q^.INFO := Q^.SIG^.INFO;`
 `P := Q;`
 `P^.SIG^.INFO := 2;`
 `WRITELN(P^.INFO , Q^.INFO);`

5) Dada la siguiente declaración

```
TYPE
  PUNTERO = ^NODO
  NODO = RECORD
    INFO : INTEGER ;
    SIG  : PUNTERO
  END;
```

Depurar los errores de las siguientes funciones o procedimientos:

a- FUNCTION PAR (P : PUNTERO) : BOOLEAN ;

```
BEGIN
  IF P^.INFO MOD 2 = 0
    THEN PAR := TRUE
    ELSE PAR := FALSE
  END;
```

b- PROCEDURE SUCEsor (P : PUNTERO) ;

```
BEGIN
  WHILE P <> NIL DO
    WRITELN(P^.INFO, ' VASEGUIDODE ', P^.SIG^.INFO);
  END ;
```

c- PROCEDURE SACAR (VAR LISTA : PUNTERO ; VAR VAL : INTEGER);

```
BEGIN
  VAL := LISTA^.INFO ;
  DISPOSE (LISTA);
  LISTA := LISTA^.SIG
END ;
```