

# CADP 2017

## Práctica 6 - Punteros

TIPO	CUÁNTO OCUPA LA REPRESENTACIÓN INTERNA EN PASCAL
INTEGER	4 bytes
REAL	8 bytes
CHAR	1 byte
STRING	Tantos bytes como indique la longitud del String + 1
RECORD	Suma de los bytes que ocupa cada campo del registro
PUNTERO	4 bytes
BOOLEAN	1 byte
ARRAY	(dimensión física del vector) x (bytes que ocupa el tipo de dato que se almacena)

1. Indique los valores que imprime el siguiente programa en Pascal.

```

program punteros;
type
  cadena = string[50];
  puntero_cadena = ^cadena;
var
  pc: puntero_cadena;
begin
  writeln(sizeof(pc), ' bytes');
  new(pc);
  writeln(sizeof(pc), ' bytes');
  pc^:= 'un nuevo nombre';
  writeln(sizeof(pc), ' bytes');
  writeln(sizeof(pc^), ' bytes');
  pc^:= 'otro nuevo nombre distinto al anterior';
  writeln(sizeof(pc^), ' bytes');
end.

```

2. Indique los valores que imprime el siguiente programa en Pascal.

```

program punteros;
type
  cadena = string[9];
  producto = record
    codigo: integer;
    descripcion: cadena;
    precio: real;
  end;
  puntero_producto = ^producto;
var
  p: puntero_producto;
  prod: producto;
begin
  writeln(sizeof(p), ' bytes');
  writeln(sizeof(prod), ' bytes');
  new(p);

```

```

    writeln(sizeof(p), ' bytes');
    p^.codigo      := 1;
    p^.descripcion := 'nuevo producto';
    writeln(sizeof(p^), ' bytes');
    p^.precio := 200;
    writeln(sizeof(p^), ' bytes');
    prod.codigo := 2;
    prod.descripcion := 'otro nuevo producto';
    writeln(sizeof(prod), ' bytes');
end.

```

3. Indique los valores que imprime el siguiente programa en Pascal.

```

program punteros;
type
    numeros = array[1..10000] of integer;
    puntero_numeros = ^numeros;
var
    n: puntero_numeros;
    num: numeros;
    i: integer;
begin
    writeln(sizeof(n), ' bytes');
    writeln(sizeof(num), ' bytes');
    new(n);
    writeln(sizeof(n^), ' bytes');
    for i:= 1 to 5000 do
        n^[i] := i;
    writeln(sizeof(n^), ' bytes');
end.

```

4. Indique los valores que imprimen los siguientes programas en Pascal.

a. **program** punteros;

```

type
    cadena = string[50];
    puntero_cadena = ^cadena;
var
    pc: puntero_cadena;
begin
    pc^:= 'un nuevo texto';
    new(pc);
    writeln(pc^);
end.

```

b. **program** punteros;

```

type
    cadena = string[50];
    puntero_cadena = ^cadena;
var
    pc: puntero_cadena;
begin
    new(pc);
    pc^:= 'un nuevo nombre';
    writeln(sizeof(pc^), ' bytes');
    writeln(pc^);
    dispose(pc);
    pc^:= 'otro nuevo nombre';
end.

```

```

C. program punteros;
  type
    cadena = string[50];
    puntero_cadena = ^cadena;
  procedure cambiarTexto(pun: puntero_cadena);
  begin
    pun^:= 'Otro texto';
  end;
  var
    pc: puntero_cadena;
  begin
    new(pc);
    pc^:= 'Un texto';
    writeln(pc^);
    cambiarTexto(pc);
    writeln(pc^);
  end.

```

```

d. program punteros;
  type
    cadena = string[50];
    puntero_cadena = ^cadena;
  procedure cambiarTexto(pun: puntero_cadena);
  begin
    new(pun);
    pun^:= 'Otro texto';
  end;
  var
    pc: puntero_cadena;
  begin
    new(pc);
    pc^:= 'Un texto';
    writeln(pc^);
    cambiarTexto(pc);
    writeln(pc^);
  end.

```

5. Indique los valores que imprimen los siguientes programas en Pascal.

```

a. program punteros;
  uses crt;
  type
    cadena = string[50];
    producto = record
      codigo: integer;
      descripcion: cadena;
      precio: real;
    end;
    puntero_producto = ^producto;

  procedure actualizar(p: puntero_producto);
  begin
    p^.descripcion := 'producto 1.1';
    p^.precio := 150;
  end;

  var
    p: puntero_producto;

```

```

begin
  writeln('Creamos un nuevo producto: ');
  new(p);
  p^.codigo := 1;
  p^.descripcion := 'producto 1';
  p^.precio := 100;

  writeln ('-----');
  writeln('Codigo:', p^.codigo);
  writeln('Descripcion:', p^.descripcion);
  writeln('Precio:', p^.precio:3:2);

  actualizar(p);

  writeln ('-----');
  writeln('Codigo:', p^.codigo);
  writeln('Descripcion:', p^.descripcion);
  writeln('Precio:', p^.precio:3:2);

  readkey;
end.

```

#### **b. program punteros;**

```
Uses crt;
```

```
type
```

```

  cadena = string[50];
  producto = record
    codigo: integer;
    descripcion: cadena;
    precio: real;
  end;

```

```
puntero_producto = ^producto;
```

```
procedure actualizar(var p: puntero_producto);
```

```
begin
```

```

  new(p);
  p^.descripcion := 'producto 1.1';
  p^.precio := 150;

```

```
end;
```

```
var
```

```
p: puntero_producto;
```

```
begin
```

```

  writeln('Creamos un nuevo producto: ');
  new(p);
  p^.codigo := 1;
  p^.descripcion := 'producto 1';
  p^.precio := 100;

```

```

  writeln ('-----');
  writeln('Codigo:', p^.codigo);
  writeln('Descripcion:', p^.descripcion);
  writeln('Precio:', p^.precio:3:2);

```

```
actualizar(p);
```

```

  writeln ('-----');
  writeln('Codigo:', p^.codigo);

```

```

        writeln('Descripcion:', p^.descripcion);
        writeln('Precio:', p^.precio:3:2);

        readkey;
    end.

```

```

C. program punteros;
Uses crt;
type
    cadena = string[50];
    producto = record
        codigo: integer;
        descripcion: cadena;
        precio: real;
    end;

    puntero_producto = ^producto;

procedure actualizar(p: puntero_producto);
begin
    new(p);
    p^.codigo := 2;
    p^.descripcion := 'producto 2';
    p^.precio := 150;
    writeln ('-----');
    writeln('Codigo:', p^.codigo);
    writeln('Descripcion:', p^.descripcion);
    writeln('Precio:', p^.precio:3:2);
end;

var
    p: puntero_producto;
begin
    writeln('Creamos un nuevo producto: ');
    new(p);
    p^.codigo := 1;
    p^.descripcion := 'producto 1';
    p^.precio := 100;

    writeln ('-----');
    writeln('Codigo:', p^.codigo);
    writeln('Descripcion:', p^.descripcion);
    writeln('Precio:', p^.precio:3:2);

    actualizar(p);

    writeln ('-----');
    writeln('Codigo:', p^.codigo);
    writeln('Descripcion:', p^.descripcion);
    writeln('Precio:', p^.precio:3:2);

    readkey;
end.

```

6. Indique los valores que imprime el programa en Pascal, si se lee la siguiente secuencia de números: **30 44 2 110 29 35 61 4 500 79 -1.**

```

program punteros;
const
    dimF = 10000;

```

```
type
  pointers = array [1..dimF] of ^integer;
var
  v: pointers;
  dimL, num: integer;
begin
  writeln(sizeof(v), ' bytes antes de cargar');
  dimL := 0;
  readln(num);
  while (num<>-1) and (dimL<dimF) do begin
    dimL := dimL+1;
    new(v[dimL]);
    v[dimL]^ := num;
    readln(num);
  end;
  writeln(sizeof(v), ' bytes después de cargar');
  writeln(sizeof(v[dimL]), ' bytes');
  writeln(sizeof(v[dimL]^), ' bytes');
end.
```