

CADP 2018

PRÁCTICA 6 – ALOCACIÓN DINÁMICA - PUNTEROS

Parte conceptual

- 1) ¿Qué es la memoria estática?
- 2) ¿Qué es la memoria dinámica?
- 3) ¿Qué es una variable del tipo puntero?
- 4) ¿Qué hace la operación NEW sobre una variable de tipo puntero?
- 5) ¿Qué hace la operación DISPOSE sobre una variable de tipo puntero?

Parte Práctica

Para los ejercicios de la parte práctica, utilizaremos la función de Pascal **sizeof**, que recibe como parámetro una variable de cualquier tipo y retorna el espacio, en bytes, que dicha variable ocupa en la memoria principal. Para realizar estos ejercicios, considere la siguiente tabla, que representa la cantidad de bytes que ocupa la representación interna de distintos tipos de datos en un compilador Pascal típico.

Tipo	Cantidad de bytes (sizeof)
Entero	4 bytes
Real	8 bytes
Char	1 byte
String	Tantos bytes como indique la longitud del String + 1
Record	La suma de las longitudes de los campos del registro
Puntero	4 bytes
Boolean	1 byte

- 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('Ocupacion en memoria de pc: ', sizeof(pc), ' bytes');
  new(pc);
  writeln('Ocupacion en memoria de pc: ', sizeof(pc), ' bytes');
  pc^:= 'un nuevo nombre';
  writeln('Ocupacion en memoria de pc: ', sizeof(pc), ' bytes');
  writeln('Ocupacion en memoria de pc^: ', sizeof(pc^), ' bytes');
  pc^:= 'otro nuevo nombre distinto al anterior';
  writeln('Ocupacion en memoria de pc^: ', 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('Ocupacion en memoria de p: ', sizeof(p), ' bytes');
  writeln('Ocupacion en memoria de prod: ', sizeof(prod), ' bytes');
  new(p);
  writeln('Ocupacion en memoria de p: ', sizeof(p), ' bytes');
  p^.codigo := 1;
  p^.descripcion := 'nuevo producto';
  writeln('Ocupacion en memoria de p^: ', sizeof(p^), ' bytes');
  p^.precio := 200;
  writeln('Ocupacion en memoria de p^: ', sizeof(p^), ' bytes');
  prod.codigo := 2;
  prod.descripcion := 'otro nuevo producto';
  writeln('Ocupacion en memoria de prod: ', 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('Ocupacion en memoria de n: ', sizeof(n), ' bytes');
  writeln('Ocupacion en memoria de num: ', sizeof(num), ' bytes');
  new(n);
  writeln('Ocupacion en memoria de n^: ', sizeof(n^), ' bytes');
  for i:= 1 to 5000 do
    n^[i] := i;
  writeln('Ocupacion en memoria de n^: ', sizeof(n^), ' bytes');
end.

```

- 4) Indique cuáles son los errores que presentan 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.

```

5) Indique qué imprimen los siguientes programas en Pascal.

```

a) program punterosA;
   type
       cadena = string[50];
       puntero_cadena = ^cadena;

   procedure cambiarTexto(var 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.

```

```

b) program punterosB;
   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.

```

```

c) program punterosC;
   type

```

```
cadena = string[50];
puntero_cadena = ^cadena;
```

```
procedure cambiar(var pun1: puntero_cadena; pun2: puntero_cadena);
begin
    pun1:= pun2;
end;
```

```
var
    p1, p2: puntero_cadena;
begin
    new(p1);
    p1^:= 'Hoy es martes';
    writeln('El contenido de p1^: ', p1^);
    cambiar(p2, p1);
    writeln('El contenido de p1^: ', p1^);
    writeln('El contenido de p2^: ', p2^);
    readln;
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.
```

e) **program** punteros;

```
type
    cadena = string[50];
    puntero_cadena = ^cadena;
procedure cambiarTexto(var 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.
```