

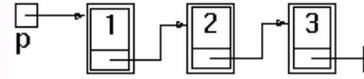
Autoevaluación del tema 11: Listas dinámicas

INF10NE0010

Enunciado

Indicar el **espacio en memoria** (en número de **bytes**) reservado para el conjunto de **variables dinámicas** que componen la lista dinámica simple de la figura, construida mediante la siguiente declaración de tipos :

```
type ptr=^elemento;
  elemento=record
    dato:longint;
    sig:ptr
  end;
```



Respuesta

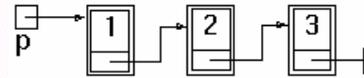
Introduce la respuesta:

INF10NE0030

Enunciado

Indicar la **salida por pantalla** tras la ejecución de la llamada al procedimiento `pr(p)`; sobre la lista dinámica mostrada en la figura de la derecha:

```
procedure pr(p:puntero);
begin
  if p^.sig<>nil then
  begin
    pr(p^.sig);
    write(p^.dato)
  end
end;
```



Respuesta

Introduce la respuesta:

INF10SS0070

Enunciado

Una estructura de datos de tipo lista dinámica en la que todas las *inserciones* o *escrituras* y *extracciones* o *lecturas* se realizan únicamente por el **mismo extremo** se denomina...

Respuesta

Marca la respuesta correcta:

1. Pila
2. Lista dinámica doblemente enlazada circular
3. Lista dinámica simple circular
4. Cola
5. Ninguna de las anteriores

INF10CO0230

Enunciado

Completar la función `ultimo` para que devuelva la **dirección de memoria** del último elemento de una **lista dinámica simple** a cuyo primer elemento apunta la variable puntero `cabeza`. En el caso de que la lista esté vacía debe devolverse el valor `nil`.

```

    Compile  Execute  Help
    [●] PREGUNTA-.PAS 1=
    type ptr=^elemento;
      elemento=record
        dato:integer;
        sig:ptr
      end;
    function ultimo(cabeza:ptr):ptr;
      var aux:ptr; { variable de uso opcional }
      begin
      { Escribe a continuación tu código }
      { Fin de tu código }
      end;
    10:1
  
```

INF10CO0240

Enunciado

Completar la función `sumalista` para que calcule y devuelva la **suma de los valores** almacenados en el campo `dato` de todos **los elementos de una lista dinámica simple** a cuyo primer elemento apunta la variable puntero `cabeza`. En el caso de que la lista esté vacía debe devolverse el valor `0`.

```

    Compile  Execute  Help
    [●] PREGUNTA-.PAS 1=
    type ptr=^elemento;
      elemento=record
        dato:integer;
        sig:ptr
      end;
    function sumalista(cabeza:ptr):integer;
      var aux:integer; { variable de uso opcional }
      begin
      { Escribe a continuación tu código }
      { Fin de tu código }
      end;
    10:1
  
```