

LISTAS

EXPLICACIÓN PRÁCTICA 7

CADP 2018

Ejercicio 1

Escriba un programa que lea y almacene información de jugadores de básquet. De cada jugador se lee: dni, apellido y nombre, y altura en cm. La lectura finaliza cuando se lee el jugador con dni 0, el cual no debe procesarse.

- ¿Cómo se representa la información de un jugador?
- ¿En qué estructura de datos se puede almacenar la información de todos los jugadores?



LISTAS – Agregar adelante

Ejemplo



Dni: 32658968
ApyNom: García Pablo
Altura: 218

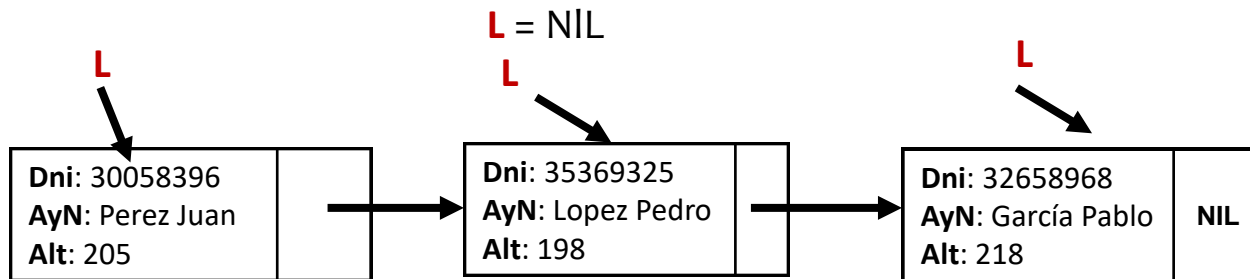


Dni: 35369325
ApyNom: Lopez Pedro
Altura: 198



Dni: 30058396
ApyNom: Perez Juan
Altura: 205

Dni: 0
ApyNom:
Altura:



Notar que quedaron almacenados en el orden inverso al leído

Solución

Ejercicio 1

```
program ejercicio;
type
  jugador = record
    dni: integer;
    nomyAp: string[30];
    altura: integer;
  end;

  lista = ^nodo;

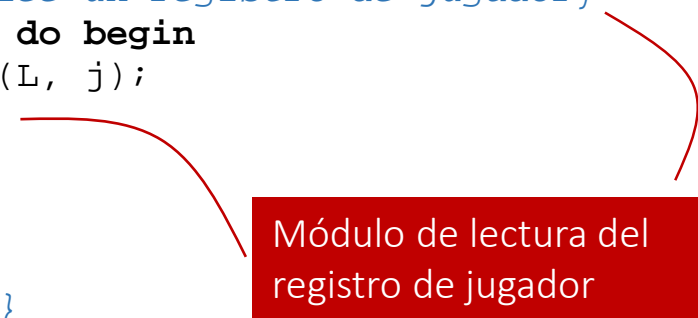
  nodo = record
    dato: jugador;
    sig : lista;
  end;
```

```
procedure agregarAdelante(var L:lista; j:jugador);
var
  nue: lista;
begin
  new (nue); {Creo un nodo}
  nue^.dato := j; {Cargo el dato}
  nue^.sig := L; {realizo el enlace}
  L:= nue; {Actualizo el primero}
end;

procedure cargarLista(var L:lista);
var
  j: jugador;
Begin
  leerJugador(j); {lee un registro de jugador}
  while(j.dni <> 0) do begin
    agregarAdelante(L, j);
    leerJugador(j);
  end;
end;

{PROGRAMA PRINCIPAL}

var
  L: lista;
begin
  L:= nil;
  cargarLista(L);
end.
```



Módulo de lectura del registro de jugador

Ejercicio 2

Modifique la solución del ejercicio 1 para que la información de los jugadores quede almacenada en el mismo orden en que fue leída.

- ¿Cómo debería realizarse la carga?



Ejemplo 2



Dni: 32658968
ApyNom: García Pablo
Altura: 218

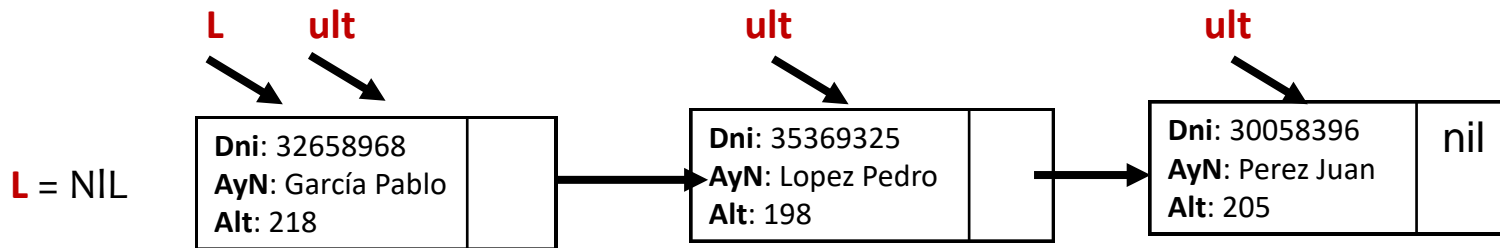


Dni: 35369325
ApyNom: Lopez Pedro
Altura: 198



Dni: 30058396
ApyNom: Perez Juan
Altura: 205

Dni: 0
ApyNom:
Altura:



Notar que quedaron almacenados en el orden leído

El puntero al primer nodo de la lista se mantiene fijo y el puntero al último se va actualizando

Solución

Ejercicio 2

```
procedure cargarLista(var L:lista);
var
  j: jugador;
  ULT: lista;
Begin
  leerJugador(j); {lee un registro de jugador}
  while(j.dni <> 0) do begin
    agregarAtras(L, ULT, j);
    leerJugador(j);
  end;
end;

procedure agregarAtras(var L, ULT:lista; j:jugador);
var
  nue: lista;
begin
  new (nue); {Creo un nodo}
  nue^.dato := j;
  nue^.sig := nil;
  if( L = nil) then
    L:= nue
  else
    ult^.sig := nue;
  ult := nue;
end;
```

Ejercicio 3

Modifique la solución del ejercicio 1 para que la información de los jugadores quede ordenada por altura de manera ascendente.

- ¿Cómo debería realizarse la carga?



LISTAS – Insertar ordenado

Ejemplo 3



Dni: 32658968
ApyNom: García Pablo
Altura: 218

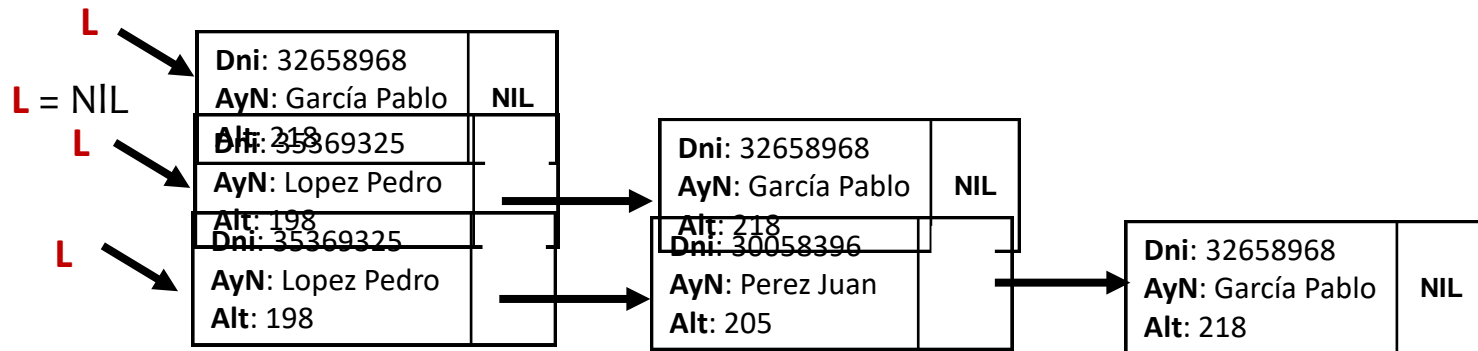


Dni: 35369325
ApyNom: Lopez Pedro
Altura: 198



Dni: 30058396
ApyNom: Perez Juan
Altura: 205

Dni: 0
ApyNom:
Altura:



Notar que quedaron almacenados ordenados por altura de forma ascendente

La forma de realizar los enlaces varía de acuerdo al lugar donde corresponde insertar el nodo

LISTAS – Insertar ordenado

Posibles casos

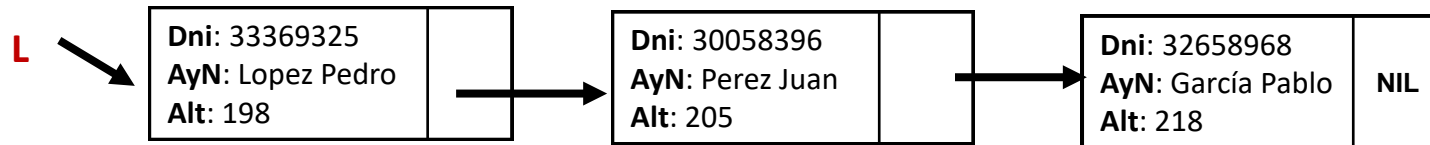
¿Cómo debe insertarse el siguiente nodo en una lista vacía ?

L = NIL

Dni: 33369325 AyN: Lopez Pedro Alt: 198	nil
--------------------------------------------------------------------	-----

Se inserta en lista vacía

Considerando ahora la siguiente lista:



- ¿Dónde corresponde insertar cada uno de los nodos que aparecen a continuación?
- ¿Cómo encuentro la posición adecuada en cada caso?
- ¿Cómo deben realizarse los enlaces?

Se inserta al inicio

Dni: 34769325 AyN: Fernández Juan Alt: 185	nil
-----------------------------------------------------------------------	-----

Se inserta al medio

Dni: 29769325 AyN: Martínez Carlos Alt: 210	nil
------------------------------------------------------------------------	-----

Se inserta al final

Dni: 32100025 AyN: González Mario Alt: 219	nil
-----------------------------------------------------------------------	-----

Pasos a seguir:

1. Crear el nodo a insertar
2. Buscar la posición correspondiente para insertar el nodo creado
 - *Es necesario ubicarse al inicio de la lista y recorrer hasta encontrar la posición.*
 - *Vamos a utilizar 2 punteros auxiliares para realizar el recorrido. ¿por qué?*
3. Realizar los enlaces
 - *Una vez encontrada la posición, deberán actualizarse los enlaces de acuerdo al caso. Es importante entonces determinar dicho caso*

Solución

Ejercicio 3

```
procedure cargarLista(var L:lista);  
var  
    j: jugador;  
Begin  
    leerJugador(j);  
    while(j.dni <> 0) do begin  
        insertarOrdenado(L, j);  
        leerJugador(j);  
    end;  
end;
```

```
procedure insertarOrdenado(var L:lista; j:jugador);  
var  
    nue: lista;  
    act, ant: lista; {punteros auxiliares para el recorrido}  
begin  
    new (nue);  
    nue^.dato := j;  
    act := L; {ubico act y ant al inicio de la lista}  
    ant := L;  
    while( act <> nil)and(j.altura < act^.dato.altura)do begin  
        ant := act;  
        act:= act^.sig;  
    end;  
    if (act = ant) then {al inicio o lista vacía}  
        L:= nue;  
    else {al medio o al final}  
        ant^.sig:= nue;  
    nue^.sig:= act;  
end;
```

1. Crear el nodo a
insertar

2. Buscar la
posición para
insertar el nodo
creado

3. Realizar los
enlaces

Ejercicio 4 – Para resolver

A partir de la lista generada en el ejercicio 1, informar la cantidad de jugadores con dni par.

