Redictado de CADP 2020

Práctica 6 – Listas

1. Dado el siguiente programa:

```
program JugamosConListas;
type
lista = ^nodo;
nodo = record
    num : integer;
    sig : lista;
end;
var
  pri : lista;
  valor : integer;
procedure armarNodo(var L: lista; v: integer);
var
  aux : lista;
begin
  new(aux);
  aux^.num := v;
  aux^.sig := L;
  L := aux;
end;
```

```
begin
  pri := nil;
  writeln('Ingrese un numero');
  read(valor);
  while (valor <> 0) then begin
    armarNodo(pri, valor);
    writeln('Ingrese un numero');
    read(valor);
  end;
  . . . { imprimir lista }
end.
```

a. Indicar qué hace el programa.

b. Indicar cómo queda conformada la lista si se lee la siguiente secuencia de números: 10 21 13 48 0.

c. Implementar un módulo que imprima los números enteros guardados en la lista generada.

d. Implementar un módulo que reciba la lista y un valor, e incremente con ese valor cada dato de la lista.

2. Dado el siguiente código, identificar los 9 errores.

```
program ejercicio2;
type
  lista = ^nodo;
  persona = record
             dni: integer;
             nombre: string;
             apellido: string;
           end;
  nodo = record
             dato: persona;
             sig: lista;
          end;

procedure leerPersona(p: persona);
begin
  read(p.dni);
  if (p.dni <> 0)then begin
    read(p.nombre);
    read(p.apellido);
  end;
end;
{Agrega un nodo a la lista}
```

```
{Carga la lista hasta que llega el dni 0}
procedure generarLista(var l:lista);
var
  p:nodo;
begin
  leerPersona(p);
  while (p.dni <> 0) do begin
    agregarAdelante(l,p);
  end;
end;

procedure imprimirInformacion(var l: lista);
begin
  while (l <> nil) do begin
    writeln('DNI: ', l^.dato.dni, 'Nombre:',
l^.nombre, 'Apellido:', l^.apellido);
    l:= l^.sig;
  end;
end;

{Programa principal}
```

```
procedure agregarAdelante(l:lista;p:persona);
var
  aux: lista;
begin
  aux^.dato:= p;
  aux^.sig:= l;
  l:= aux;
end;
```

```
var
  l:lista;
begin
  generarLista(l);
  imprimirInformacion(l);
end.
```

3. Utilizando el programa del ejercicio 1, realizar los siguientes cambios:
a. Modificar el módulo *armarNodo* para que los elementos se guarden en la lista en el orden en que fueron ingresados (agregar atrás).
b. Modificar el módulo *armarNodo* para que los elementos se guarden en la lista en el orden en que fueron ingresados, manteniendo un puntero al último ingresado.

4. Utilizando el programa del ejercicio 1, realizar los siguientes módulos:
a. *Máximo*: recibe la lista como parámetro y retorna el elemento de valor máximo.
b. *Mínimo*: recibe la lista como parámetro y retorna el elemento de valor mínimo.
c. *Múltiplos*: recibe como parámetros la lista L y un valor entero A, y retorna la cantidad de elementos de la lista que son múltiplos de A.

5. Realizar un programa que lea y almacene la información de productos de un supermercado. De cada producto se lee: código, descripción, stock actual, stock mínimo y precio. La lectura finaliza cuando se ingresa el código -1, que no debe procesarse. Una vez leída y almacenada toda la información, calcular e informar:
a. Porcentaje de productos con stock actual por debajo de su stock mínimo.
b. Descripción de aquellos productos con código compuesto por al menos tres dígitos pares.
c. Código de los dos productos más económicos.

6. La Agencia Espacial Europea (ESA) está realizando un relevamiento de todas las sondas espaciales lanzadas al espacio en la última década. De cada sonda se conoce su nombre, duración estimada de la misión (cantidad de meses que permanecerá activa), el costo de construcción, el costo de mantenimiento mensual y rango del espectro electromagnético sobre el que realizará estudios. Dicho rango se divide en 7 categorías:

1. radio; 2. microondas; 3.infrarrojo; 4. luz visible; 5. ultravioleta; 6. rayos X; 7. rayos gamma

Realizar un programa que lea y almacene la información de todas las sondas espaciales. La lectura finaliza al ingresar la sonda llamada "GAIA", que debe procesarse.
Una vez finalizada la lectura, informar:
a. El nombre de la sonda más costosa (considerando su costo de construcción y de mantenimiento).
b. La cantidad de sondas que realizarán estudios en cada rango del espectro electromagnético.
c. La cantidad de sondas cuya duración estimada supera la duración promedio de todas las sondas.
d. El nombre de las sondas cuyo costo de construcción supera el costo promedio entre todas las sondas.
Nota: para resolver los incisos a), b), c) y d), la lista debe recorrerse una única vez.

7. El Programa Horizonte 2020 (H2020) de la Unión Europea ha publicado los criterios para financiar proyectos de investigación avanzada. Para los proyectos de sondas espaciales vistos en el ejercicio anterior, se han determinado los siguientes criterios:
- sólo se financiarán proyectos cuyo costo de mantenimiento no supere el costo de construcción.
- no se financiarán proyectos espaciales que analicen ondas de radio, ya que esto puede realizarse desde la superficie terrestre con grandes antenas.

A partir de la información disponible de las sondas espaciales (la lista generada en ejercicio 6), implementar un programa que:
a. Invoque un módulo que reciba la información de una sonda espacial, y retorne si cumple o no con los nuevos criterios H2020.

b. Utilizando el módulo desarrollado en a) implemente un módulo que procese la lista de sondas de la ESA y retorne dos listados, uno con los proyectos que cumplen con los nuevos criterios y otro con aquellos que no los cumplen.
c. Invoque a un módulo que reciba una lista de proyectos de sondas espaciales e informe la cantidad y el costo total (construcción y mantenimiento) de los proyectos que no serán financiados por H2020. Para ello, utilice el módulo realizado en b.

8. Utilizando el programa del ejercicio 1, modificar el módulo *armarNodo* para que los elementos de la lista queden ordenados de manera ascendente (insertar ordenado).

9. Utilizando el programa del ejercicio 1, realizar los siguientes módulos:
a. *EstáOrdenada*: recibe la lista como parámetro y retorna *true* si la misma se encuentra ordenada, o *false* en caso contrario.
b. *Eliminar*: recibe como parámetros la lista y un valor entero, y elimina dicho valor de la lista (si existe). Nota: la lista podría no estar ordenada.
c. *Sublista*: recibe como parámetros la lista L y dos valores enteros A y B, y retorna una nueva lista con todos los elementos de la lista L mayores que A y menores que B.
d. Modifique el módulo Sublista del inciso anterior, suponiendo que la lista L se encuentra ordenada de manera ascendente.
e. Modifique el módulo Sublista del inciso anterior, suponiendo que la lista L se encuentra ordenada de manera descendente.

10. Una empresa de sistemas está desarrollando un *software* para organizar listas de espera de clientes. Su funcionamiento es muy sencillo: cuando un cliente ingresa al local, se registra su DNI y se le entrega un número (que es el siguiente al último número entregado). El cliente quedará esperando a ser llamado por su número, en cuyo caso sale de la lista de espera. Se pide:
a. Definir una estructura de datos apropiada para representar la lista de espera de clientes.
b. Implementar el módulo *RecibirCliente*, que recibe como parámetro el DNI del cliente y la lista de clientes en espera, asigna un número al cliente y retorna la lista de espera actualizada.
c. Implementar el módulo *AtenderCliente*, que recibe como parámetro la lista de clientes en espera, y retorna el número y DNI del cliente a ser atendido y la lista actualizada. El cliente atendido debe eliminarse de la lista de espera.
d. Implementar un programa que simule la atención de los clientes. En dicho programa, primero llegarán todos los clientes juntos, se les dará un número de espera a cada uno de ellos, y luego se los atenderá de a uno por vez. El ingreso de clientes se realiza hasta que se lee el DNI 0, que no debe procesarse.

11. La Facultad de Informática debe seleccionar los 10 egresados con mejor promedio a los que la UNLP les entregará el premio Joaquín V. González. De cada egresado se conoce su número de alumno, apellido y el promedio obtenido durante toda su carrera.
Implementar un programa que:
a. Lea la información de los todos egresados, hasta ingresar el código 0, el cual no debe procesarse.
b. Una vez ingresada la información de los egresados, informe el apellido y número de alumno de los egresados que recibirán el premio. La información debe imprimirse ordenada según el promedio del egresado (de mayor a menor).

12. Una empresa desarrolladora de juegos para teléfonos celulares con Android dispone de información de todos los dispositivos que poseen sus juegos instalados. De cada dispositivo se conoce la versión de Android instalada, el tamaño de la pantalla (en pulgadas) y la cantidad de memoria RAM que posee (medida en GB). La información disponible se encuentra ordenada por versión de Android. Realizar un programa que procese la información disponible de todos los dispositivos e informe:
a. La cantidad de dispositivos para cada versión de Android.
b. La cantidad de dispositivos con más de 3 GB de memoria y pantallas de a lo sumo a 5 pulgadas.
c. El tamaño promedio de las pantallas de todos los dispositivos.

13. El Portal de Revistas de la UNLP está estudiando el uso de sus sistemas de edición electrónica por parte de los usuarios. Para ello, se dispone de información sobre los 3600 usuarios que utilizan el portal. De cada usuario se conoce su email, su rol (1: Editor; 2. Autor; 3. Revisor; 4. Lector), revista en la que participa y cantidad de días desde el último acceso.
a. Imprimir el nombre de usuario y la cantidad de días desde el último acceso de todos los usuarios de la revista *Económica*. El listado debe ordenarse a partir de la cantidad de días desde el último acceso (orden ascendente).
b. Informar la cantidad de usuarios por cada rol para todas las revistas del portal.
c. Informar los emails de los dos usuarios que hace más tiempo que no ingresan al portal.

14. La oficina de becas y subsidios desea optimizar los distintos tipos de ayuda financiera que se brinda a alumnos de la UNLP. Para ello, esta oficina cuenta con un registro detallado de todos los viajes realizados por una muestra de 1300 alumnos durante el mes de marzo. De cada viaje se conoce el código de alumno (entre 1 y 1300), día del mes, Facultad a la que pertenece y medio de transporte (1. colectivo urbano; 2. colectivo interurbano; 3. tren universitario; 4. tren Roca; 5. bicicleta). Tener en cuenta que un alumno puede utilizar más de un medio de transporte en un mismo día.
Además, esta oficina cuenta con una tabla con información sobre el precio de cada tipo de viaje.
Realizar un programa que lea la información de los viajes de los alumnos y los almacene en una estructura de datos apropiada. La lectura finaliza al ingresarse el código de alumno -1, que no debe procesarse.
Una vez finalizada la lectura, informar:
a. La cantidad de alumnos que realizan más de 6 viajes por día
b. La cantidad de alumnos que gastan en transporte más de $80 por día
c. Los dos medios de transporte más utilizados.
d. La cantidad de alumnos que combinan bicicleta con algún otro medio de transporte.

EJERCICIOS ADICIONALES

- La cátedra de CADP está organizando la cursada para el año 2019. Para ello, dispone de una lista con todos los alumnos que cursaron EPA. De cada alumno se conoce su DNI, apellido, nombre y la nota obtenida. Escribir un programa que procese la información de alumnos disponible y los distribuya en turnos utilizando los siguientes criterios:
  - Los alumnos que obtuvieron al menos 8 en EPA deberán ir a los turnos 1 ó 4.
  - Los alumnos que obtuvieron entre 5 y 8 deberán ir a los turnos 2, 3 ó 5.
  - Los alumnos que no alcanzaron la nota 5 no se les asignará turno en CADP.

  Al finalizar, el programa debe imprimir en pantalla la lista de alumnos para cada turno.
  Nota: La distribución de alumnos debe ser lo más equitativa posible.

- La empresa distribuidora de una app móvil para corredores ha organizado una competencia mundial, en la que corredores de todos los países participantes salen a correr en el mismo momento en distintos puntos del planeta. La app registra, para cada corredor, el nombre y apellido, la distancia recorrida (en kilómetros), el tiempo (en minutos) durante el que corrió, el país y la ciudad desde donde partió, y la ciudad donde finalizó su recorrido. Realizar un programa que permita leer y almacenar toda la información registrada durante la competencia. Una vez que se han almacenado todos los datos, informar:
  - La cantidad total de corredores, la distancia total recorrida y el tiempo total de carrera de todos los corredores.
  - El nombre de la ciudad que convocó la mayor cantidad de corredores y la cantidad total de kilómetros recorridos por los corredores de esa ciudad.
  - La distancia promedio recorrida por corredores de Brasil.
  - La cantidad de corredores que partieron de una ciudad y finalizaron en otra ciudad.
  - El paso (cantidad de minutos por km) promedio de los corredores de la ciudad de Boston.