Práctica Nº 9

Tema: Estructuras de datos dinámicas. Listas simplemente encadenadas (LSE). Listas doblemente encadenadas (LDE).
Duración: 4 clases
Esta práctica tiene como objetivos
Conocer y emplear el tipo puntero para la creación y manipulación de estructuras de datos dinámicas.
Introducir el concepto de secuencia, para el tratamiento de datos dinámicos de manera secuencial.
Conocer y emplear estructuras dinámicas más complejas como las doblemente encadenas.
Reflexionar sobre la diferencia del empleo de estructuras de datos dinámicas y estáticas en el diseño de algoritmos.

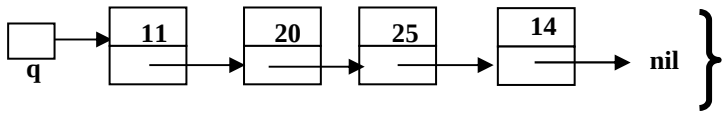
Ejercicios propuestos

Ej. 1) Dada una lista simplemente encadenada definida mediante TNodo, y las variables de tipo puntero **r**, **t**, **s** y **q**.

a) Describe gráficamente los estados intermedios y el estado final a partir del estado inicial y del segmento de algoritmo dados mas abajo.

```
Algoritmo Ej1
Lexico
  TNodo= <info ∈  entero, next ∈  puntero a TNodo>
  q, r, t, s, p: puntero a TNodo
Inicio
  ... {creación de la lista simplemente encadenada}
```

eo:




```
  r ← q
  r ← (^r).next
  t ← (^r).next
  Obtener(s)
  (^s).info ← 3
  (^s).next ← (^r).next
  (^r).next ← s
  (^t).info ← 29
  ... {mostrar los componentes de la lista simplemente encadenada}

Fin
```

b) Escribe en pseudolenguaje la parte del algoritmo {creación de la lista simplemente encadenada}, necesario para alcanzar el estado eo.

c) Escribe el pseudolenguaje al final del algoritmo (mostrar los componentes de la lista simplemente encadenada) que nos permita visualizar por pantalla todos los elementos de la lista.

d) Traduce el algoritmo y la acción, de puntos anteriores, a C.

Ej.2) Desarrolla acciones que permitan manipular una **lista de números enteros** de la siguiente manera:

a) Acción InsertarC, que permita insertar un elemento al comienzo de la lista.
b) Acción InsertarF, que permita insertar un elemento al final de la lista
c) Acción Inicializar, que permita crear la lista vacia o vaciarla si tiene elementos.

d) Acción InsertarPos, que permita insertar una elemento en una posición dada (si esta existe).
e) Acción EliminarC, que permita eliminar el primer elemento de la lista.
f) Acción EliminarPos, que permita eliminar un elemento de una posición dada, si esta existe.
g) Acción Mostrar, que permita visualizar en pantalla los elementos de la lista.
h) Desarrollar un algoritmo que mediante un menú de opciones, permita acceder a cada una de las acciones antes enumeradas. Agregar una opción para terminar.
i) Traduce a C las acciones resultantes e implemente en computadora.

Ej.3) Desarrolla acciones que permitan manipular una **lista de personas** de la siguiente manera:

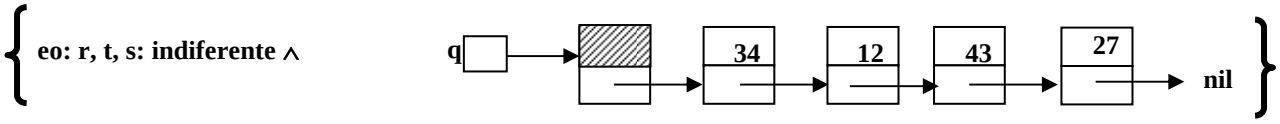
a) Insertar un elemento al comienzo de la secuencia.
b) Eliminar el primer elemento de la secuencia.
c) Imprimir por pantalla todos los elementos (personas) de la secuencia.
d) LeerReg, para cargar los datos de una persona. La acción debe leer los campos de un registro TPersona. Esta acción debe ser invocada antes de la acción insertar, a la cual se pasará el registro cargado como un parámetro dato.
e) Desarrollar un algoritmo que mediante un menú de opciones, permita acceder a cada una de las acciones antes enumeradas. Agregar una opción para terminar.
f) Traduce a C el algoritmo resultante e implemente en computadora.

El tipo persona se define como sigue:

En notación algorítmica
TPersona = <nom ∈ cadena, edad ∈ [0..120], fechNac ∈ cadena>

Ej. 4) Dada una lista simplemente encadenada con elemento ficticio, y las variables de tipo puntero **r**, **t**, **s** y **q**.

a) Describe gráficamente los estados intermedios y el estado final a partir del siguiente estado inicial, que se produce el siguiente segmento de algoritmo.




```
r ← (^q).next
t ←(^r).next
(^q).next ←t
(^r).next ← (^t).next
(^t).next ← r
r ← (^r).next
Obtener(s)
(^s).info ← 9
(^s).next ← (^r).next
(^r).next ← s
```

Piensa, ¿Cómo se crea una lista con elemento ficticio?

b) Traduce el segmento de algoritmo a C.

Ej.5) Desarrolle acciones, utilizando punteros y **elemento ficticio**, que permitan manipular una **lista** de la siguiente manera:

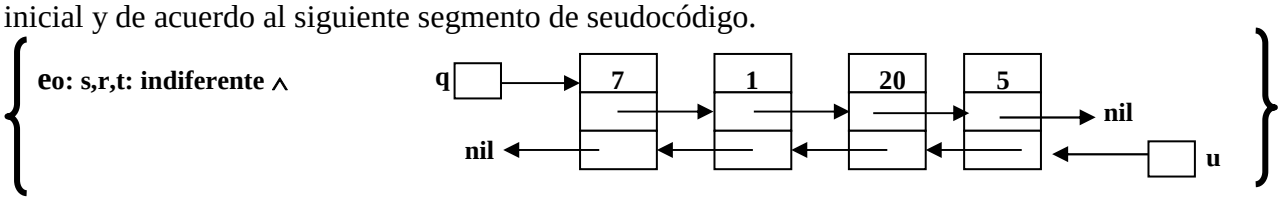
a) Acción InsertarC, que permita insertar un elemento al comienzo de la secuencia.
b) Acción InsertarF, que permita insertar un elemento al final de la secuencia
c) Acción Inicializar, que permita crear la secuencia vacia o vaciarla si tiene elementos.

d) Acción InsertarPos, que permita insertar una elemento en una posición dada (si esta existe).
e) Acción EliminarPos, que permita eliminar un elemento de una posición dada, si esta existe.
f) Desarrolla una función que permita BUSCAR un empleado por el nombre. Esta función debe devolver un entero que representa la posición en la secuencia de la primera aparición de una persona con ese nombre. En caso de no existir, la función debe devolver –1.
g) Acción MuestraReg que permita visualizar los datos de un empleado en pantalla. Al invocar esta acción debe previamente utilizar la función BUSCAR y en caso que el empleado exista, pasar como parámetro a MuestraReg la posición del empleado cuyas datos van a ser visualizados.
h) Acción Mostrar, que permita visualizar en pantalla los elementos de la secuencia.
i) Desarrollar un algoritmo que mediante un menú de opciones, permita acceder a cada una de las acciones antes enumeradas. Agregar una opción para terminar.
j) Traduce a C las acciones resultantes e implemente en computadora.

El tipo empleado se define como sigue:

En notación algorítmica
TEmpleado = <nombre ∈ Cadena, telefono ∈ Cadena, direccion ∈ Cadena, edad ∈ [18..65]>

Ej. 6) Dada una lista doblemente encadenada y las variables de tipo puntero **s**, **r**, **t** y **q**. Describe gráficamente los estados intermedios y el estado final partiendo del siguiente estado inicial y de acuerdo al siguiente segmento de seudocódigo.




```
s ← q
s ← (^s).next
Obtener(r)
Obtener(t)
(^r).info ← 8
(^r).back ← s
(^r).next ← (^s).next
(^(^s).next).back ← r
(^s).next ← r
s ← (^s).next
(^t).info ← 12
(^t).back ← s
(^t).next ← (^s).next
(^(^s).next).back ← t
(^s).next ← t
```

b) Traduce el segmento de algoritmo a C.

Ej.7) Dada una lista doblemente encadenada que implementa una **lista de al menos diez números enteros**, desarrolla acciones que permitan:
a) Insertar un entero, que se recibe como parámetro, en una posición dada.
b) Ahora resuelve el inciso a) considerando que tiene un **elemento ficticio**.
c) Imprimir los elementos de la lista desde el final al principio.

Ej. 8) Dada una LSE de números enteros que implementa una secuencia, desarrolle una **función** que cuente cuántos elementos tiene la lista (ese es lo que devuelve).

Ej. 9) Desarrolla una función de tipo lógica que devuelva verdadero si la LSE que se pasa cómo parámetro está vacía y sino que devuelva falso. Realiza dos versiones: una para LSE con elemento ficticio y otra para lista sin elemento ficticio.

Ej. 10) Dada un LSE de números enteros que implementa una secuencia, ordenados de menor a mayor, desarrolle una acción que inserte un número (que será pasado cómo parámetro al igual que el puntero cabeza de la LSE) y mantenga la LSE ordenada.

Plan de Clases

Clase 1: ejercicio 1) y 2) traer hechos lo que falte
Clase 2: ejercicio 4) , 5)
Clase 3: 6), 7)
Clase 4: 8), 9) y 10)

Para entregar en C:

Desarrolle un algoritmo con el siguiente menú de opciones:

a) Inicializar una lista (crea una lista vacía)
b) Insertar un elemento en la lista (debe insertar al final)
c) Suprimir un elemento de la lista (debe suprimir a la cabeza de la lista)
d) Mostrar (debe visualizar todos los elementos de la lista)
e) Salir

Debe desarrollar cada una de las acciones y parametrizar adecuadamente. El tipo de los elementos debe ser un registro o tipo compuesto con no menos de dos campos.

Vencimiento: **en la fecha que indique el profesor a cargo de los Trabajos Prácticos**