Universidad Católica del Norte  
Facultad de Ingeniería y Ciencias Geológicas Departamento de Ingeniería de Sistemas y Computación

Puntaje:

**Nota**:

ESTRUCTURA DE DATOS – CÁTEDRA 1 (II-2024)

Nombre - RUT: Fecha: 13 de Noviembre del 2024.

***Competencias a evaluar:*** Programación de estructuras de datos básicas utilizando listas enlazadas.

1. Usando Stack y/o Queue (la cantidad que estime necesaria), Imprimir los datos de un arreglo en el orden que se muestra. Asuma que todas las casillas del arreglo ya contienen la información. Por simplicidad del ejercicio, “n” siempre es par. (60 puntos)

void imprimir(int\* A, int n)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1° | 3° | 5° |  | (n-3)° | (n-1)° | 2° | 4° |  | (n-4)° | (n-2)° | n° |
| ? | ? | ? | … | ? | ? | ? | ? | … | ? | ? | ? |
| 0 | 1 | 2 |  | (n/2) – 2 | (n/2) - 1 | n/2 | (n/2) + 1 |  | n-3 | n-2 | n-1 |

Ejemplo:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 17 | 1 | 5 | -3 | 0 | -21 | -9 | 37 | 6 | -12 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Al imprimir este conjunto de datos, la secuencia esperada es: 17, -21, 1, -9, 5, 37, -3, 6, 0, -12

Observaciones:

* Los datos vienen inicialmente en un arreglo A listo para ser usado.
* No se permite imprimir los datos directamente desde el arreglo.
* Se DEBE traspasar los datos a un Stack o una Queue y posterior a este traspaso (no antes) se deben imprimir los datos solicitados en el orden señalado.

1. Considere la siguiente lista generalizada resolviendo lo que se les solicite:

A = ( C , a , B , ( C , c ) , b )

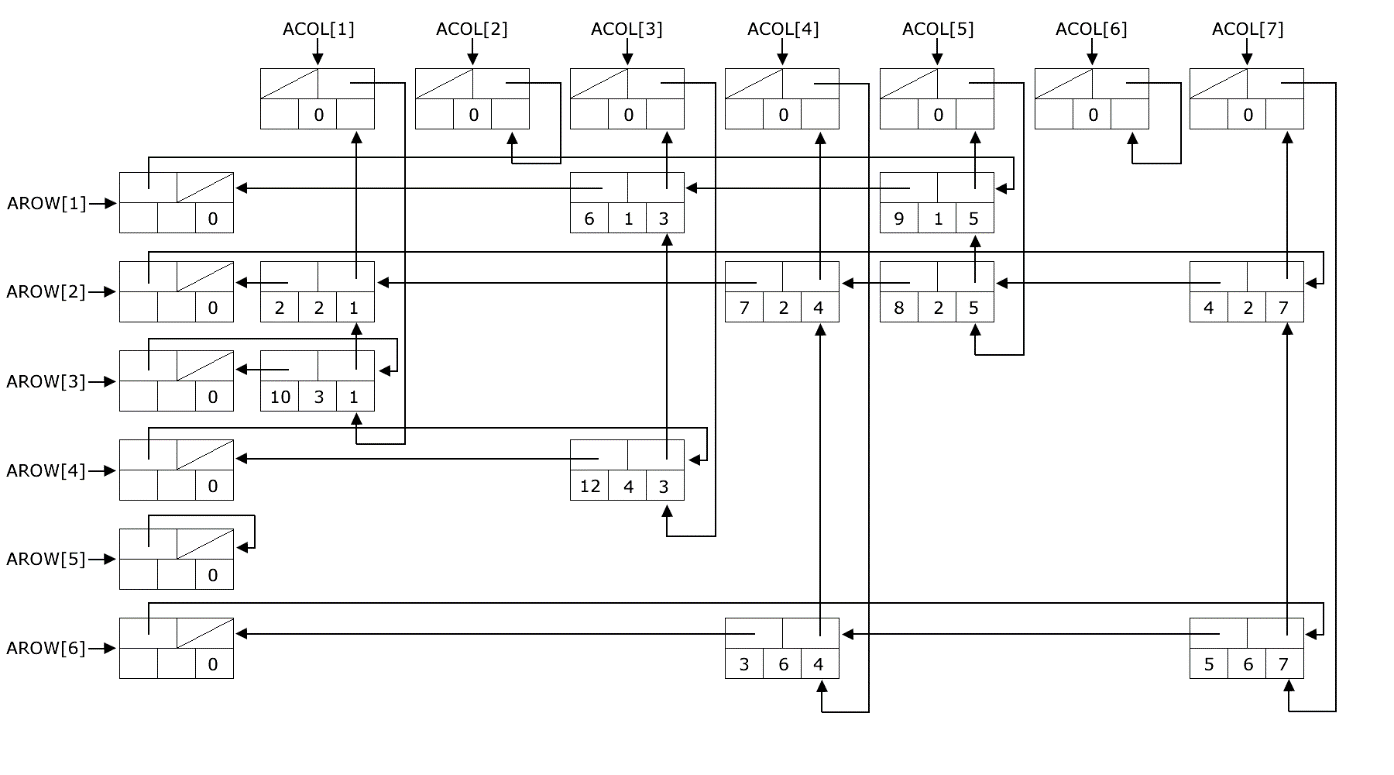
B = ( e , ( A , ( f ) , ( ) ) , ( ( ) , ( ( ) , d ) ) )

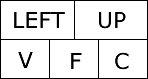
C = ( B , ( h , ( ( c ) ) ) )

* 1. Mostrar la Lista generalizada gráficamente con el formato de Nodos visto en clases. (40 puntos)
  2. Si se eliminan todas las referencias a la sublista B, señale cómo serian todos los accesos posibles al átomo “c” mediante únicamente operaciones Head() y Tail().   
     (20 puntos)
     1. Las eliminaciones señaladas implican borrar el elemento como un todo, es decir, no queda un espacio vacío posterior a la eliminación.
     2. En el apartado anterior no deben de reflejarse estas eliminaciones.
     3. Tampoco se pide representar gráficamente la lista generalizada resultante al realizar las eliminaciones.

1. En una Matriz poco poblada (Mpp) de dimensiones NxM, elabore un código que señale por pantalla cuál de las **columnas** (número) contiene la mayor cantidad de nodos en **filas** impares. El formato del código que se pide es el siguiente: (60 puntos)

void columna\_mas\_nodos(MPP\* matriz, int n, int m);





Formato del Nodo:

Observaciones:

* Las imágenes son solo guías de apoyo a los ejercicios, no corresponden a la forma exacta de los datos con los que se trabaja. Su código debe funcionar para cualquier conjunto de datos.
* En ningún ejercicio se pide elaborar el código de alguna clase ni el código del programa principal (main). Solo se pide implementar una función que cumpla con lo que se solicita en el enunciado de cada ejercicio.
  + Aunque se reciba código correspondiente a clases como tal o el programa “main”, estos no tendrán puntaje asociado. Esto independiente de si están bien o mal implementados.
* Debido a la naturaleza del ejercicio 2, se permiten entregas en formato mezclado, es decir, algunos ejercicios en digital y otros en formato lápiz y papel. La elección queda a criterio de cada estudiante.
* No se pide la entrega de ningún proyecto, solo el código de las funciones señaladas. Estas pueden ser entregadas en formato “.txt”, “.docx”, o similares. Puede preguntarle al profesor para corroborar si así lo desea.

**Ejercicio 1**

Texto

Descripción generada automáticamente

Crear las estructuras necesarias (2): 10 puntos

Rellenar las estructuras con los datos del arreglo: 20 puntos

Imprimir los datos respetando el orden señalado: 30 puntos

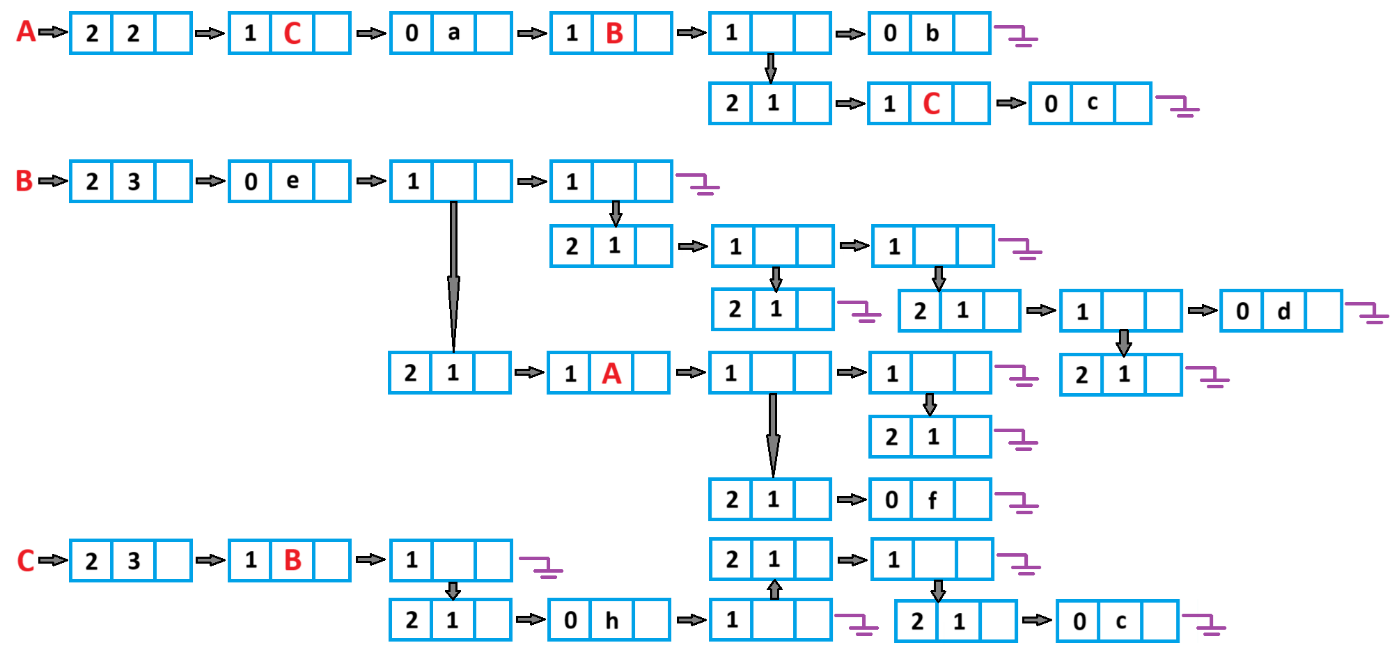
Cualquier puntaje parcial obtenido será señalado y explicado en comentarios dentro del desglose de puntaje obtenido por cada estudiante.

**Ejercicio 2.a)**

A = ( C , a , B , ( C , c ) , b )

B = ( e , ( A , ( f ) , ( ) ) , ( ( ) , ( ( ) , d ) ) )

C = ( B , ( h , ( ( c ) ) ) )



**Ejercicio 2.b)**

A = ( C , a , ( C , c ) , b )

B = ( e , ( A , ( f ) , ( ) ) , ( ( ) , ( ( ) , d ) ) )

C = ( ( h , ( ( c ) ) ) )

1. H(T(H(T(T(A))))) **🡪 acceso abreviado 1 = HTHTT = α**
2. H(H(H(T(H(C))))) **🡪 acceso abreviado 2 = HHHTH = β**

Accediendo a “A” desde “B”

1. H(H(T(B))) = A 🡪 **α(**H(H(T(B)))**)**

Accediendo a “C” desde “A”

1. H(A) = C 🡪 **β(**H(A)**)**
2. H(H(T(T(A)))) = C 🡪 **β(**H(H(T(T(A))))**)**

Accediendo a “C” desde “B” pasando por “A”

1. H(H(T(B))) = A 🡪 acceso 4 desde A hasta C

🡪 **H(**H(H(T(B)))**)** = C 🡪 **β(H(**H(H(T(B)))**))**

1. H(H(T(B))) = A 🡪 acceso 5 desde A hasta C

🡪 **H(H(T(T(**H(H(T(B)))**))))** = C 🡪 **β(H(H(T(T(**H(H(T(B)))**)))))**

Para efectos prácticos del ejercicio, se solicitó un mínimo de 4 accesos mientras que en total existen 7. Independiente del acceso que se entregue dentro de la prueba como una respuesta, se otorgará puntaje total o parcial en cada caso siempre considerando que se evalúan solo 4 de los accesos.

**Ejercicio 3**

Texto

Descripción generada automáticamente

Variables inicializadas: 8 puntos

Ciclo For o equivalente con acceso a la cabecera: 16 puntos

Ciclo while avanzando correctamente: 16 puntos

Contar filas impares correctamente por columna: 8 puntos

Determinar e imprimir valor mayor solicitado: 12 puntos