

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**ALGORITMIA**

**Laboratorio 3**

**2015-1**

**Indicaciones generales:**

- Duración: 2h 50 min.
- Al inicio de cada respuesta, el alumno deberá incluir, a modo de comentario, la estrategia que utilizará para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en la pregunta correspondiente.
- Si la implementación es significativamente diferente a la estrategia indicada o no la incluye, la pregunta será corregida sobre el 50 % del puntaje asignado y sin derecho a reclamo.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 60 % del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado.
- El orden será parte de la evaluación.
- Su trabajo deberá ser subido a PAIDEIA en el espacio indicado por los Jefes de Práctica.

---

**Pregunta 1 (6 puntos)**

Dado un número entero positivo  $N$  entre 1 y 500, usando programación dinámica, imprima la menor cantidad necesaria de cuadrados perfectos cuya suma sea igual a  $N$ . Además imprima dichos cuadrados.

La entrada contiene varias líneas. Cada línea es un caso de prueba y contiene el número  $N$  para el cual se debe ejecutar el algoritmo. La última línea contiene un número negativo, que indica el fin de los casos de prueba.

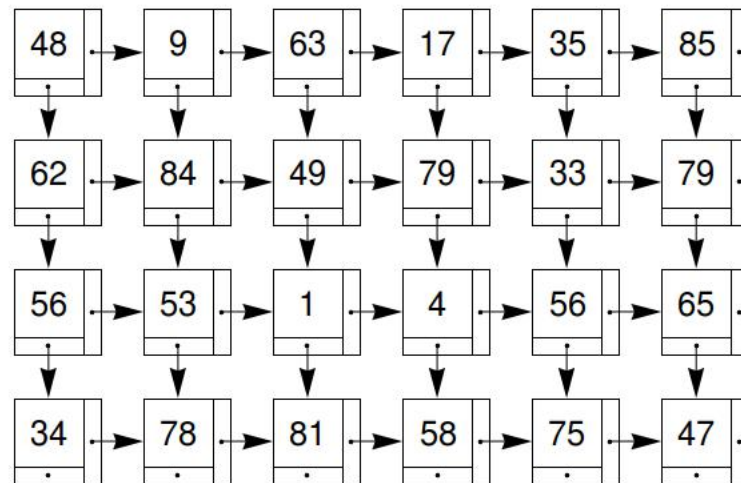
Entrada	Salida
13	2 4 + 9
432	3 16 + 16 + 400
79	4 1 + 4 + 25 + 49
-1	

## Pregunta 2 (3 puntos)

Dado un nodo de una lista simplemente enlazada, se desea eliminar dicho nodo de la lista en tiempo  $O(1)$  a pesar de no conocer el inicio de la lista ni el nodo anterior al que se desea eliminar. Implemente la función `deleteNode(ListNode* node)` que reciba un nodo de una lista de enteros simplemente enlazada y lo elimine de la lista. Puede asumir que el nodo que recibirá no es ni el primero ni el último de la lista, pero la operación debe realizarse en tiempo  $O(1)$ . Implemente la función en el archivo `p2.c` descargado de PAIDEIA.

## Pregunta 3 (11 puntos)

Se desea implementar una estructura de datos (Matrix) similar a una matriz de enteros, pero donde cada elemento es un nodo similar al de una lista enlazada. Cada nodo contiene dos referencias a nodos siguientes (derecha y abajo), como se muestra a continuación.



Defina los tipos de datos necesarios (2 puntos) e implemente las funciones:

- `void Matrix_print(Matrix* matrix)` (1 punto)  
Imprime los elementos de *matrix*, de manera que la *i*-ésima línea impresa contiene los elementos de la *i*-ésima fila de *matrix*.
- `void Matrix_addColumn(Matrix* matrix, List column)` (1.5 puntos)  
Agrega los elementos de la lista simplemente enlazada *column* como una columna nueva de *matrix*. La cantidad de elementos de *column* debe coincidir con la cantidad de filas de *matrix* a menos que *matrix* esté vacía.
- `void Matrix_addRow(Matrix* matrix, List row)` (1.5 puntos)  
Agrega los elementos de la lista simplemente enlazada *row* como una fila nueva de *matrix*. La cantidad de elementos de *row* debe coincidir con la cantidad de columnas de *matrix* a menos que *matrix* esté vacía.
- `void Matrix_deleteColumn(Matrix* matrix, int i)` (1.5 puntos)  
Elimina la *i*-ésima columna de *matrix*.
- `void Matrix_deleteRow(Matrix* matrix, int i)` (1.5 puntos)  
Elimina la *i*-ésima fila de *matrix*.

- void Matrix\_transpose(Matrix matrix, Matrix\* transpose) **(2 puntos)**  
Crea la matriz transpuesta de *matrix* y guarda el resultado en *transpose*.

Su función *main* debe incluir llamadas a las funciones solicitadas de manera que se pueda comprobar que las funciones implementadas están funcionando correctamente.

Profesores del curso: Robert Ormeño  
Fernando Alva

Pando, 26 de mayo del 2015