

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Laboratorio 3

2016-2

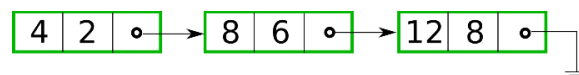
Indicaciones generales:

- Duración: 2h 50 min.
 - Al inicio de cada programa, 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 esa pregunta.
 - 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 (10 puntos)

Un arreglo **esparsa** es un arreglo de números enteros con una muy pequeña cantidad de elementos diferentes de cero. En aplicaciones reales, los arreglos esparsa podrían llegar a tener tamaños de millones de elementos, de los cuales muy pocos son diferentes de cero. Como es de esperarse, crear un arreglo para almacenar sólo algunos elementos hace que desperdiciemos espacio almacenando los números ceros.

Una representación alternativa de un arreglo esparsa es usar listas enlazadas, en donde cada nodo almacena una posición y el número que va en esa posición. Por ejemplo, el arreglo $A=\{0,0,0,0,2,0,0,0,6,0,0,8,0,0,0\}$ de tamaño 16 sólo tiene elementos en las posiciones 4, 8, 12; por lo que se podría representar como una lista enlazada como



El primer nodo indica que en la posición 4 hay número 2. El segundo nodo indica que en la posición 8 hay un número 6. El tercer nodo indica que en la posición 12 hay un número 8. Observe que la lista se encuentra ordenada con respecto a la posición.

El objetivo de esta pregunta es implementar una función en C que sume dos arreglos esparsa del mismo tamaño usando la representación de listas enlazadas.

Entrada

La primera línea contiene un solo número n que indica el tamaño de los arreglos esparsa. A continuación se especifica un arreglo en cada línea de la siguiente forma: cada par de números indica la posición del elemento y el elemento en sí.

Salida

Una sola línea que contiene la lista resultante en el mismo formato que la entrada.

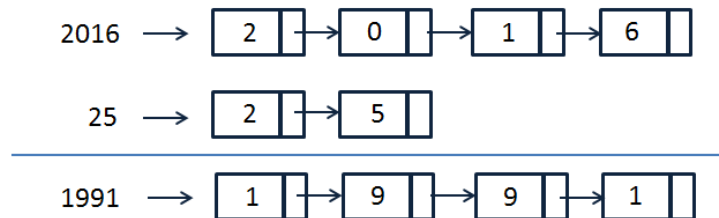
Ejemplo:

Entrada	Salida
16 4 2 8 6 12 8 1 4 4 5 6 8 12 9	1 4 4 7 6 8 8 6 12 17

- El primer arreglo contiene en la posición 4 un número 2, en la posición 8 un número 6 y en la posición 12 un número 8.
- El segundo arreglo contiene en la posición 1 un número 4, en la posición 4 un número 5, en la posición 6 un número 8 y en la posición 12 un número 9.

Pregunta 2 (10 puntos)

Una técnica para representar enteros largos (números que exceden el rango de valores que pueden ser representados) es usar listas enlazadas en las cuales cada nodo contiene un dígito del número. Dadas 2 listas enlazadas que representan 2 números enteros positivos largos, **restar** el número menor del mayor y retornar la diferencia en una lista enlazada. Nótese que las listas de entrada pueden estar en cualquier orden, pero siempre se debe necesitar restar el menor del mayor.



En el ejemplo, 2016 es representado por una lista de 4 nodos, donde cada nodo contiene un dígito del número. La diferencia con 25 es 1991, que es representado también por una lista de 4 nodos.

Entrada

La primera línea contiene el número de casos de prueba a testar. Para cada caso de prueba se tienen 2 líneas que representan los números a restar.

Salida

Una sola línea que contiene la diferencia de los 2 números en una lista enlazada.

Ejemplo:

Entrada	Salida
2	
7654321753574 45452	7654321708122
7654321753573 7654321753574	1

Nota: Para convertir un char a int puede usar lo siguiente:

```
char t = 'a';  
int b = t - '0';
```

Profesores del curso: Marco Sobrevilla

Ivan Sipiran

Pando, 25 de Octubre del 2016