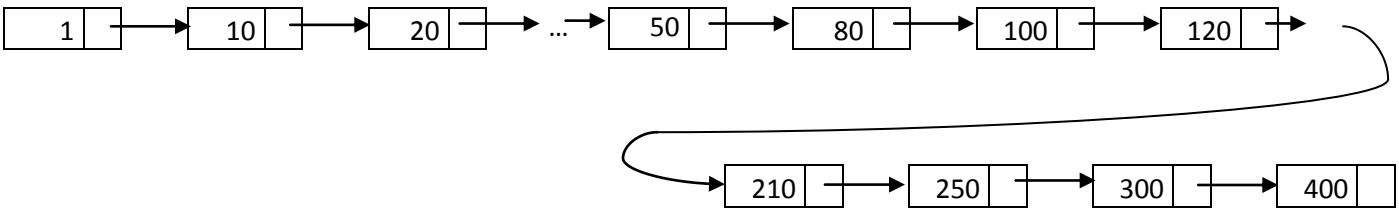
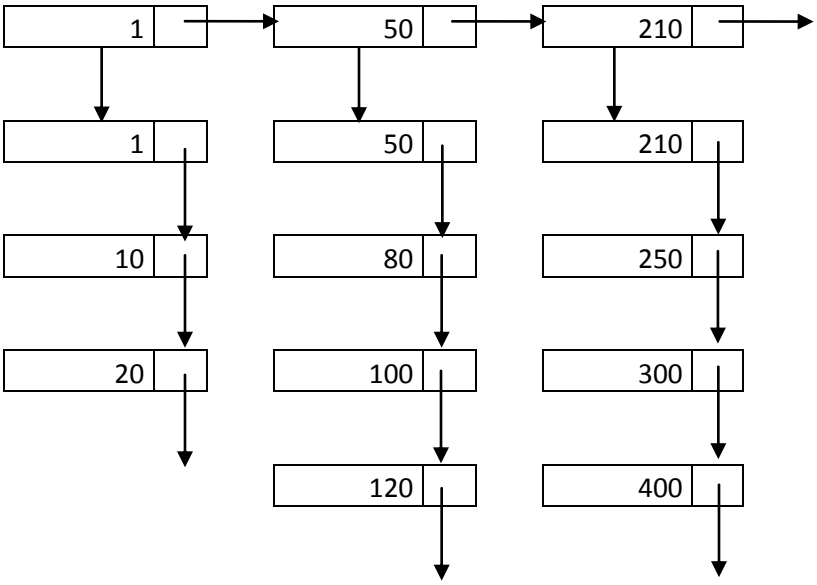
Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Programación de Computadores - Fundamentos de Programación
Segundo Parcial Conjunto - Octubre 17 de 2014

1. (30%) Elabore una función recursiva que muestre cuántos y cuáles números del NUM1 al NUM2 no tienen cifras repetidas (NUM1 > 99 y NUM2 < 1000).

2. (40%) Cuando una lista ordenada tiene un tamaño considerable, una forma más eficiente de representarla es creando una secuencia encadenada de nodos, en donde cada uno de ellos tiene una entrada a cierto número de elementos de la lista. Por ejemplo la Lista:



Se puede representar así:



a. (10%) Defina los tipos de datos necesarios para manejar una lista ordenada eficiente con las características mencionadas.

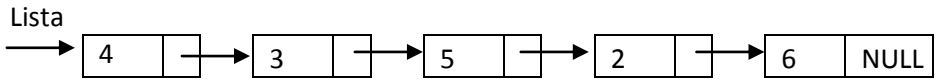
b. (30%) Realice una función adicionarElemento que reciba como parámetro el código del elemento a adicionar teniendo en cuenta lo siguiente:
   1. Los nodos de entrada deben tener acceso a mínimo 3 elementos de la lista
   2. Los nodos de entrada deben dar acceso a máximo 10 elementos de la lista, es decir que si el nuevo elemento se debe adicionar a uno que ya tiene 10 elementos se debe reorganizar la lista para que esta condición se siga cumpliendo.

3. (30%) Dada una lista cuya definición de tipos es la siguiente:

```
typedef  int  TElemento;

typedef struct nodo
{
   TElemento elem;
   struct nodo *sig;
 } TNodo;
```

Se pretende realizar una función que, mediante un algoritmo recursivo, calcule la suma de los enteros almacenados en los nodos cuyo valor sea superior a un valor umbral que se pasará como parámetro a la función. Ejemplo: A partir de la siguiente lista y el valor umbral 3.



La suma devuelta por la función es 15 que corresponde a la suma de 4, 5 y 6 que son los números mayores de 3 que hay en la lista. Se pide codificar en C/C++, la función cuyo prototipo es:

```
void sumaEnteros (TNodo *lista, int umbral, int *suma);
```

Nota: suma es un parámetro de salida.