Lebanese University
Faculty of Sciences 1

INFO216

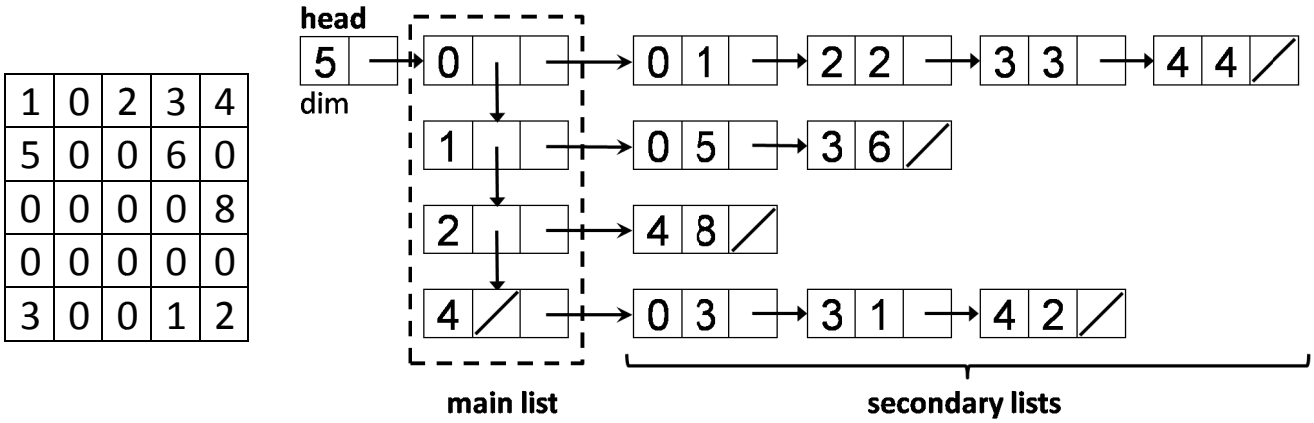
Imperative Programming II

Final 08.09 (1 July 09)
Duration: 2 hours

A sparse matrix is a matrix with lots of zeros. We represent sparse matrices using linked lists of heads of linked lists of non-zero elements of the matrix (compact form):

- each element of the main list contains the index of the line with at least one non-zero element and a pointer to a secondary list for the elements of this line,
- each element of a secondary list represents a non-zero element in the matrix (column index and value),
- the matrices we are dealing with are **square**,
- the head of the main list contains a field called 'dim' to indicate the dimension of the square matrix.

Normal Form **Compact Form**



1. Define the adequate data types to be used to represent a sparse matrix in its compact form.
2. Write the function "construct" that builds dynamically the compact form representation of a given sparse matrix (2D array).
3. Write the function "add" that calculates and returns the sum of two sparse matrices in their compact form. Remember to verify dimension compatibility. The result of the addition must also be in the compact form.
4. Write the function "show" that displays a compact sparse matrix in its normal form.
5. Write the function "save" that writes a compact sparse matrix into a file. This function must also liberate the memory reserved by the matrix.
6. Continue the function main() below by adding the appropriate calls to the functions written in 1 to 5.

```
void main(){
   int A[5][5]= {1,0,2,3,4,5,0,0,6,0,0,0,0,0,8,0,0,0,0,0,3,0,0,1,2};
   int B[5][5]= {0,8,0,0,0,0,0,0,0,5,0,3,0,0,2,0,1,1,0,2,0,0,0,1,0};
   //a- declare the variables matA, matB and matR
   //b- construct matA and matB : the compact forms of A and B (resp.).
   //c- add matA and matB and put the result in matR
   //d- display matR
   //e- save matR into a file called "answer"
}
```

The End