Birla Institute of Technology & Science, Pilani
Data Structures & Algorithms (CS F211)
Lab Assignment – 9 A (Graph Representation)

Instructions:

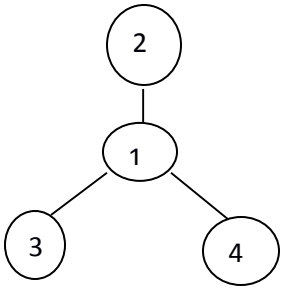
- All input expressions should be read from stdin and output should be printed on stdout.
- For 1 hour 45 min, only a subset of test cases will be visible to students after submitting the code on the portal. After 1 hour 45 min, all test cases will be made visible and they will have last 15 min to correct their code and resubmit.
- At the end of 2 hour period, the online system will stop accepting the submissions.
- Only the last submission by the student before end of lab will be considered for evaluation.
- Following messages by online portal will **tentatively** fetch these marks:
  - Correct ➔ 4 marks
  - Wrong-answer (correct for more than half test cases) ➔ 3 marks
  - Run-error/Compiler-error/Timelimit-error ➔ 2 marks
- All submitted source code will be later checked manually by the instructor and final marks will be awarded, which will be posted on Nalanda after the lab assignment has been done by all lab sections.
- Solution must be implemented using the algorithm and data structures mentioned in the lab sheet only.
- Do not delete your code right away, you will need it for Part-B of this assignment.

Problem

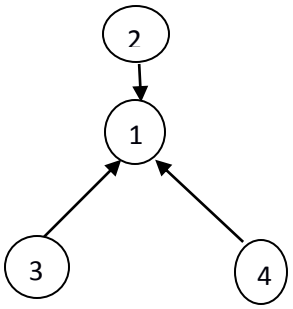
Methods of graph representation and their interchangibility.

Input

First line of the input will be a code type XYZ, where X∈{D,U} is the type of graph. D for directed graph and U for undirected graph. Y∈{A,I,L} is the *input* graph representation. Z∈{A,I,L}is the *output* graph representation. A stands for adjacency matrix, I for incidence matrix and L for adjacency list representation. Please refer "lecture 20" and "lecture 21" on Nalanda for different graph representations. After the first line of code "XYZ", from the second line you will be given "Y" representation of a graph. Your task is to output the "Z" representation of the input graph. The formats for the three graph representations (for both input and output) are as follows:



G1



G2

1. Adjacency Matrix ("A") Representation:

First line will have the number of vertices n. From second line onwards rows of the adjacency matrix will be printed, one row per line, and the values will be separated by blanks. For example, "A" representation of G1 will be:

4
0 1 1 1
1 0 0 0
1 0 0 0
1 0 0 0

Vertices will always be numbered in sequential order stating from 1. The matrix should be in sorted order of vertices.

2. Adjacency List ("L") Representation:

First line will have the number of vertices n. From second line onwards adjacency vertices of each vertex will be listed per row in the following format:

vertex number blank number of adjacency vertices blank first adjacency vertex blank second adjacency vertex blank …….

All entries will be in serial order. For example, "L" representation of G1 will be:

4
1 3 2 3 4
2 1 1
3 1 1
4 1 1

3. Incidence Matrix ("I") Representation:

First line will have the number of vertices n. Second line will have the number of edges e. From third line onwards rows of the incidence matrix will be printed, one row per line, and the values will be separated by blanks. For example, "I" representation of G2 will be:

4

3

1 1 1

-1 0 0

0 -1 0

0 0 -1

Edge numbering convention: Sort the edges (i, j) on i. For "tie: breaking use j. For G2, the sorted edges are:

<2,1>,<3,1>,<4,1>

Sample Input 1

ULI

4

1 3 2 3 4

2 1 1

3 1 1

4 1 1

Sample Output 1

4

3

1 1 1

1 0 0

0 1 0

0 0 1

Sample Input 2

DAL

4

0 0 0 0

1 0 0 0

1 0 0 0

1 0 0 0

Sample Output 2

4

1 0

2 1 1

3 1 1

4 1 1