

- [Assessments](#)
- [Statistics](#)
- [Machine Problems 1](#)

Yuetong Liu

[Log out](#)

#1.1 MP1

Purpose:

Get deeper understanding of BCNF decomposition algorithms through implementation.

Implementation:

- You can use Java/Python as your programming language. No other languages are allowed.
- You can use standard libraries but you are prohibited to include external libraries.
- You are required to **strictly follow the input/output formats**, since we will use **auto-graders** to grade your submissions.
- Your code needs to read its input from an input file and write the output to an output file.
- The input and output files will be passed to your program on the command line (The first argument is the input file name, and the second one is the output file name). A sample command to run your program:
`java MP1 arg1 arg2`
- This is an individual assignment, which means it is OK to discuss with your classmates and TAs regarding the methods, but it is not OK to work together or share code.
- Copying source code from other students will give you 0 grade. We will run plagiarism detection software.

Submission:

- Your source code file should be named as either MP1.java or MP1.py
- Put all your code in a folder with the name MP1 and compress it into a zip file and name it as MP1.zip. Do not use sub-folders inside this folder. All of your code should have been successfully compiled before submission. Do not include files other than the code you write. Put a single readme.txt file in the code folder to briefly describe the functionalities of your code.
- Submit your MP1.zip file through the upload link at the bottom of the page.

Description:

In this MP you are required to implement BCNF Decomposition. Your code will get a relation R as the input and should determine whether it is in BCNF or not. If R is not in BCNF, you need to decompose the relation (based on BCNF decomposition) and output the obtained relations. Your implementation can be broken into two parts: a module to check whether a given relation is in BCNF and another module for decomposition based on BCNF. In the following, we describe the format of the input and output files.

Input:

You will get a relation as the input of your code. A relation can be described by attribute names and dependencies between them. The first line of the input file contains attribute names separated by comma

character. The second line contains an integer n , which denotes the number of dependencies. The next n lines each represents a dependency, containing two sets of features separated by “->”, denoting the features coming before “->” functionally determine the features coming after “->”. The feature names in each of the two sets are separated by comma character.

Output:

If the input relation is in BCNF form, print “YES” (without quotes) in the first line. Otherwise, print “NO” which means the input relation is not in BCNF. The second line should represent the number of relations obtained from decomposition, followed by the description of each relation. To describe each relation follow the same format as we used to represent the input relation: the first line contains feature names, the second line contains an integer n denoting the number of the dependencies that hold in the relation and the next n lines each represents a dependency. It is evident that if the input schema is in BCNF you should not decompose it. In such a case, your output file should have a single line which contains “YES”. If there are multiple ways to decompose a relation, you may print any of them.

Sample input/output files can be downloaded from [here](#).

Upload your MP1.zip file

No file chosen

Machine Problems 1

Score: 0%

Question #1.1

Attempts: 0

Score: 0%

[Previous question](#)

[Next question](#)