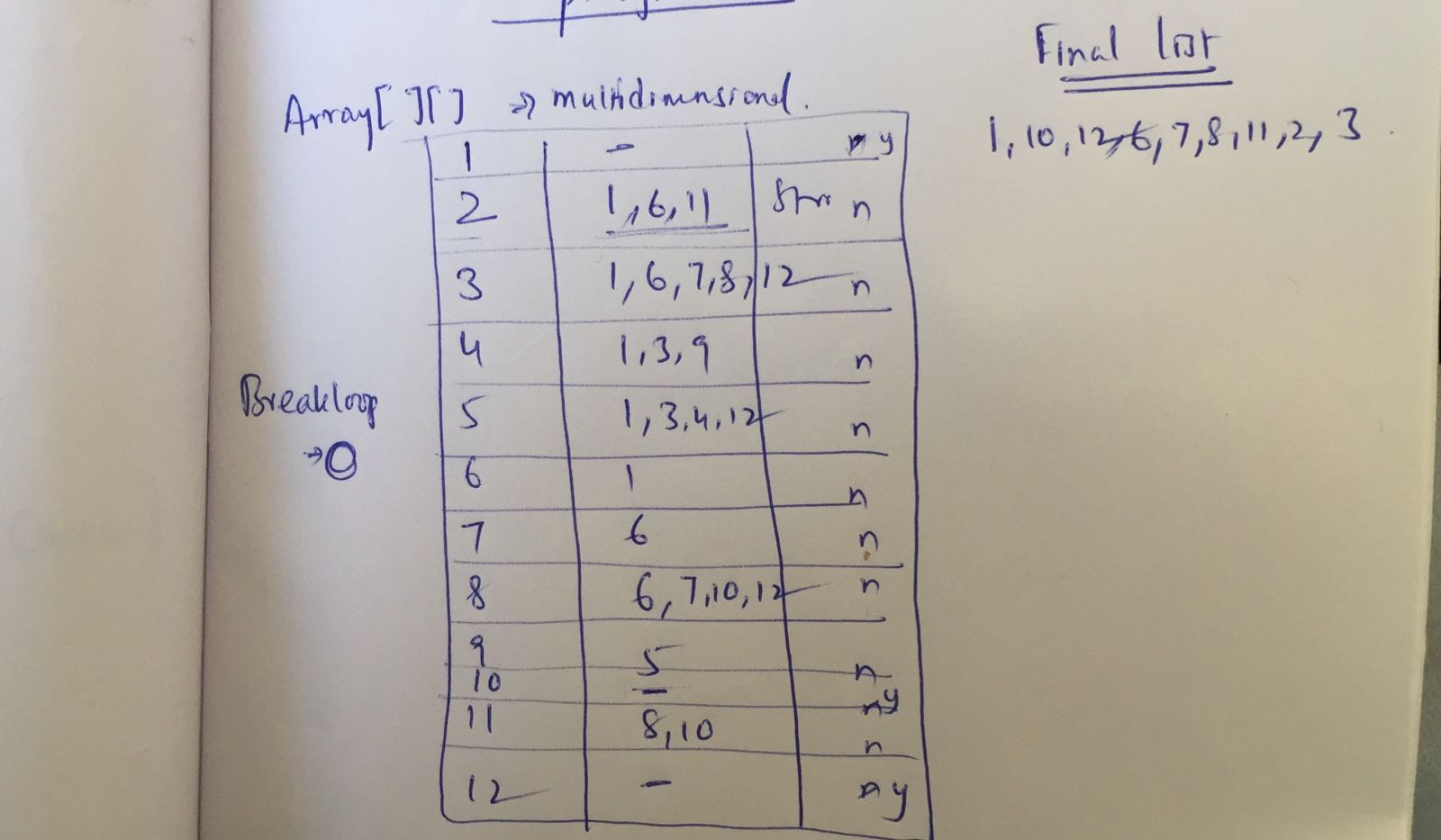
In order to solve the problem given to me, I have used BlueJ (a java platform) for which I feel very comfortable in executing my ideas that gives best results. As I have created a New project, my plan is to read csv files and fetch the data from them. To do that I have copied the files into the project folder and loaded into the program.

First, I have created a two-dimensional array to make the data from course and prerequisite csv into arrays. Next, I have filled all the empty values of array by default. As I have read the course.csv file data into an array, I have combined all the prerequisites into one and separated with ‘,’ delimiter. Then I have created an ArrayList named ‘FinalList’ which will contain all the courses in an order that helps student to complete his enrolment.

The algorithm I used to solve this problem is that at first, all the courses which are not having any prerequisite units are given a flag ‘yes’ and all the courses with flag yes are directly sent into the final list. Now, the courses with flag ‘no’ will be compared with the courses that are already in the final list. For all the comparisons made in the loop, if only all the courses from the final list will be matched with the prerequisite unit(s), that specific course will be sent into final list and the loop continues to find all the courses for matched prerequisites.

The image below gives a clue on How my algorithm actually works.



**Challenges:**

1. As the units 4,5,9 have dependent prerequisites among themselves, an infinite loop is formed and to break this loop, I have used a variable named BreakLoop which is initially 0, if no unit enters the FinalList in the iteration, the BreakLoop will be changed to 1 thereby breaking the loop. **And because of the interdependencies one can only enrol in 9 units.** This may lead to incompletion of the course so changing those dependencies may help in completing the course.

If any changes are made for the prerequisites in the csv, the program is designed in such a way that it can choose the order dynamically.

**Execution:**

In order to compile and run my program in linux workstation the following are the commands to be used.

Change path to the current directory and open the “challenge” folder then,

Compiling – *javac test.java*

Run – *java test*