Your friend is writing a new compiled programming language called Smurf. He believes that it will be one of the best programming languages ever. He has almost finished implementing the compiler but is struggling with one last issue.

In Smurf, programmers can write their codes into multiple files and can import other files or libraries. Before the compiler compiles a file, the files that the file to be compiled depends on must be already compiled. Therefore, the compiler should know the compilation order of files. Also, for some programs, compilation may not be possible because of cyclic dependencies (an example of cyclic dependency, two programs may be trying to import each other). In the case of cyclic dependency, the compiler should warn the user by giving the sets of files that cause the cyclic dependency. You offer to help your friend by writing a function to help this compiling process.

Problem

In this exam, you are expected to implement the *run* function shown below. If the program you are trying to compile is compilable, you should assign true to the isCompilable variable, and assign the order of compilation shown by the file IDs to the compileOrder variable. If it is not compilable due to cyclic dependencies, you should assign the IDs of cyclic dependent files to the cyclicDependencies variable, a vector including vectors for each cyclic-dependent file group.

```
void run(const std::vector<std::vector<int>>& dependencyMatrix, bool&
isCompilable,

        std::vector<int>& compileOrder, std::vector<std::vector<int>>&
cyclicDependencies)
```

- *dependencyMatrix*: Square matrix with dimensions FxF, where F is the number of files. If dependencyMatrix[i][j] is 1, then file *j is dependent on file i*. Otherwise, it is 0.
- *isCompilable*: If compilation is possible, you should assign true, otherwise false.
- compileOrder: If compilation is possible, you should assign compilation order to this variable.
- cyclicDependencies: If compilation is not possible, you should find cyclic-dependent file groups and update this vector to include the file groups by their IDs. *Cyclic-dependent file group is a group of files that each file in the group is directly or indirectly dependent on every other file in the group.*

Constraints and Hints:

- If the compilation order does not matter for two files, you should sort them via indexes. (Smaller index, higher priority)

- Self-import is also possible. You should check them. Also, there may be indirect dependencies.(i.e 1 -> 2 -> 3 -> 1)
- The return order of cyclic-dependent files is not important and each file can be an element of at most one cyclic-dependent file group.
- The number of files is less than 100.