Submission deadline (on or before):

- 20.10.2020, 10:00 PM

Policies for Submission and Evaluation:

- You must submit your assignment in the Eduserver course page, on or before the submission deadline.
- Your programs should be written in C language and should be compatible with the `gcc` compiler in Linux.
- During the evaluation, failure to execute programs without compilation errors may lead to zero marks for the evaluation.
- Detection of ANY malpractice related to the lab course can lead to awarding an F grade in the course.

Naming Conventions for Submission

- Submit a single ZIP (.zip) file containing the source file (do not submit in any other archived formats like .rar, .tar, .gz). The name of this file must be named as

  `DCS<NUMBER>_<ROLLNO>_<FIRST-NAME>.zip`

  (For example: *DCS1_BxxyyyyCS_LAXMAN.zip*). DO NOT add any other files (like temporary files, input files, etc.) or folders, except your source code, into the zip archive.

- The source codes must be named as

  `DCS<NUMBER>_<ROLLNO>_<FIRST-NAME>_<PROGRAM-NUMBER>.c`

  (For example: *DCS1_BxxyyyyCS_LAXMAN_1.c*). If you do not conform to the above naming conventions, your submission might not be recognized by our automated tools, and hence will lead to a score of 0 marks for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

- Violation of academic integrity will be severely penalized. Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course. The department policy on academic integrity can be found at: http://cse.nitc.ac.in/sites/default/files/Academic-Integrity_new.pdf.

1. Write a program to color a given planar undirected graph G using one of the 6-colors from the set $\{a, b, c, d, e, f\}$, as specified below.

Input Format

The first line of the input contains an integer $n \in [1, 10^4]$, the number of vertices in G. The vertices are implicitly labeled from 0 to $n - 1$.

The second line of the input contains an integer $m \in [0, 3 \times 10^4]$, the number of edges in G.

The next m lines of the input each contain two space-separated integers, say $u, v \in [0, n - 1]$, indicating the presence of the undirected edge (u, v) in G.

Output Format

The output should contain n lines, corresponding to each of the n vertices of G. Each of these lines should contain a single character from the set $\{a, b, c, d, e, f\}$, indicating the color assigned by the algorithm to the corresponding vertex.

Algorithm Outline

1. If G contains only one vertex, assign is color a.
2. Otherwise, pick a vertex, say v, with degree at most 5 from $V(G)$ and recursively 6-color the graph $G \setminus v$.
3. After returning from the recursion, assign v a color that none of its neighbors have been assigned.

Implementation Details

The algorithm outlined above can produce different colorings based on the vertex picked (in step 2) and the color assigned (in step 3). As such, the following conventions must also be followed to ensure that the colors assigned by the algorithm are uniquely determined.

1. In step 2, the algorithm must pick a vertex with the smallest label from the set of all eligible vertices.
2. In step 3, the algorithm must assign the possible smallest color (assume that the colors are ordered alphabetically, with a being the smallest and f being the largest).

As this is not a course on programming, you are *not* discouraged from using global variables. Feel free to use them to simplify your implementation. The prototypes of the functions you use may also be decided accordingly.

Sample Input and Output

Input

```
7
9
0 4
1 2
```

```
1 3
1 4
2 4
3 4
4 5
4 6
5 6
```

Corresponding Output

```
a
b
a
a
c
b
a
```