Problem # 7 – Car license plates

Write a program which interprets an input data sequence consisting of one or more lines. The program determines whether each line is a valid Romanian registration number (for a vehicle), utilizing a set of rules, in which case it displays those lines on the standard output.

Entry data

An input sequence consists of one or more lines, each one terminated by *newline* (\n), generated by pressing *Enter*. Each line consists of three strings (of characters) separated by a space character. The structure of each line is as follows:

String1 String2 String3

where String1, String2, and String3 are strings whose structure will be described in the next section.

The internal logic of the program

The program will verify that, taken together, the three strings from each line represent a valid registration number, according to the following rules:

1. the valid values for String1 are: AB, AR, AG, B, BC, BH, BN, BT, BV, BR, BZ, CS, CL, CJ, CT, CV, DB, DJ, GL, GR, GJ, HR, HD, IL, IS, IF, MM, MH, MS, NT, OT, PH, SM, SJ, SB, SV, TR, TM, TL, VS, VL, VN (Please note: only uppercase!);
2. String2 consists of 2 or 3 numeric characters (there is no other restriction in connection to String1);
3. String3 consists of exactly 3 uppercase characters.

Output data

The program must output (on the *stdout* stream) only the input lines representing a valid registration number under the rules above. The lines containing valid numbers will not be altered in any way, and their order will be preserved. Output lines are terminated by the *newline* character (\n).

WARNING to the compliance of the problem requirement: displaying the results should be EXACTLY on the indicated way! In other words, the standard output stream will not show anything in addition to the requirements of the problem; as a result of automatic evaluation, any additional character displayed, or viewed other than that indicated, will lead to a false result and therefore obtain a Rejection of the program.

Restrictions and specifications

- **Warning**: Depending on the chosen programming language, the file containing the code must have one of the following extensions .c, .cpp, .java, or .m. The web editor **will not automatically add an extension** and its absence leads to the impossibility of compiling the program!
- **Warning**: The source file must be named, by the candidate, in the following form: <name>.<ext>, where name is the surname of the candidate and the extension is chosen according to the previous point. Pay attention to the limitations of the Java language, related to the class name and file name!

Examples

Input text	Output text	Explanations
AB 123 ABC B 23 DEF	AB 123 ABC B 23 DEF	Both input lines (license plates), presented here, are correct, therefore we have exactly the same output as the input.
BB 123 ABC SV 99 DEF	SV 99 DEF	The line (license plate) beginning with BB is invalid.
B 1234 ABC BV 9 ABC VN 01 ABC	VN 01 ABC	The first line is invalid because the associated String2 has 4 characters. The second line is invalid because the associated String2 has only 1 character.
AB 11 AAA CT 0A XYZ CV 01 AB8	AB 11 AAA	The second line has an illegal value of the String2. The third line has an illegal value of the String3.
B 01 abc		The program does not display anything because the received number on input is wrong. String3 contains small letters.

Working time: 120 minutes