Your Informatics teacher at school likes coming up with new ways to help you understand the material. When you started studying numeral systems, he introduced his own numeral system, which he's convinced will help clarify things. His numeral system has base 26, and its digits are represented by English capital letters - A for 0, B for 1, and so on.

The teacher assigned you the following numeral system exercise: given a one-digit number, you should find all unordered pairs of one-digit numbers whose values add up to the number.

Example

For number = 'G', the output should be  
newNumeralSystem(number) = ["A + G", "B + F", "C + E", "D + D"].

Translating this into the decimal numeral system we get: number = 6, so it is ["0 + 6", "1 + 5", "2 + 4", "3 + 3"].

Input/Output

* **[execution time limit] 3 seconds (java)**
* **[input] char number**

A character representing a correct one-digit number in the new numeral system.

*Guaranteed constraints:*  
'A' ≤ number ≤ 'Z'.

* **[output] array.string**
  + An array of strings in the format "letter1 + letter2", where "letter1" and "letter2" are correct one-digit numbers in the new numeral system. The strings should be sorted by "letter1".

*Note* that "letter1 + letter2" and "letter2 + letter1" are equal pairs and we don't consider them to be different.

**[Java] Syntax Tips**

// Prints help message to the console

// Returns a string

//

// Globals declared here will cause a compilation error,

// declare variables inside the function instead!

String **helloWorld**(String name) {

System.out.println("This prints to the console when you Run Tests");

**return** "Hello, " + name;

}

<https://app.codesignal.com/challenge/oHHb2WWoMkYSGEPbx>

static String[] newNumeralSystem(char number) {

char i = 'A';

char j = number;

ArrayList<String> vec = new ArrayList();

while(i <= j) {

vec.add(String.valueOf(i) + " + " + String.valueOf(j));

i++;

j--;

}

return vec.toArray(new String[0]);

}