A *squared encoding* in a simple encoding algorithm that can be used to encrypt messages of moderate importance. To encrypt a `message` consisting of lowercase English letters, the following steps should be taken:

1. consider `message`characters one by one;
2. each letter is assigned to a number: `'a'` to `1`, `'b'` to `2`, and so on, with `'y'`assigned to `25` and `'z'`assigned to `0`;
3. take a value assigned to the character, calculate its square modulo `26`, and add the letter assigned to the obtained result to the answer.

Given a `message`, your task is to encode it using the *squared encoding* algorithm.

Example

For `message = "hello"`, the output should be

`decode2(message) = "lynnq"`.

Here's why:

- `'h' -> 8`^{2} `% 26 = 64 % 26 = 12 -> 'l'`;
- `'e' -> 5`^{2} `% 26 = 25 % 26 = 25 -> 'y'`;
- `'l' -> 12`^{2} `% 26 = 144 % 26 = 14 -> 'n'`;
- `'o' -> 15`^{2} `% 26 = 225 % 26 = 17 -> 'q'`.

Input/Output

- **[time limit] 4000ms (py3)**
- **[input] string message**

  A string consisting of lowercase English letters.

  Constraints:

  `1 ≤ message.length ≤ 100`.

- **[output] string**

  Squared encoded `message`.