A. Recovering a Small String

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Nikita had a word consisting of exactly 3 lowercase Latin letters. The letters in the Latin alphabet are numbered from 1 to 26, where the letter "a" has the index 1, and the letter "z" has the index 26.

He encoded this word as the sum of the positions of all the characters in the alphabet. For example, the word "cat" he would encode as the integer $3 + 1 + 20 = 24$, because the letter "c" has the index 3 in the alphabet, the letter "a" has the index 1, and the letter "t" has the index 20.

However, this encoding turned out to be ambiguous! For example, when encoding the word "ava", the integer $1 + 22 + 1 = 24$ is also obtained.

Determine the lexicographically smallest word of 3 letters that could have been encoded.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \neq b$;
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 100$) — the number of test cases in the test.

This is followed by the descriptions of the test cases.

The first and only line of each test case contains an integer n ($3 \leq n \leq 78$) — the encoded word.

Output

For each test case, output the lexicographically smallest three-letter word that could have been encoded on a separate line.

Example

input Copy

```
5
24
70
3
55
48
```

output Copy

```
aav
rzz
aaa
czz
auz
```