

7053 NAND

Xiaoqiang entered the “shortest code” challenge organized by some self-claimed astrologists. He was given a boolean function taking n inputs (in C++):

```
bool f(bool x1, bool x2, bool x3){
//your code goes here
//return something
}
```

All possible inputs and expected outputs of this function have been revealed (see table on the right):

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Xiaoqiang’s code must be like:

```
bool a = NAND(b,c);
```

where “ a ” is a newly defined variable, “ b ” and “ c ” can be a constant (0/1) or a function parameter ($x_1/x_2/x_3$) or a previously defined variable. NAND is the “not-and” function:

```
NAND(b,c)=!(b&&c)
```

Because NAND is universal, Xiaoqiang knew that he could implement any boolean function he liked. Also, at the end of the code there should be a return statement:

```
return y;
```

where y can be a constant or a function parameter or a previously defined variable. After staring at the function for a while, Xiaoqiang came up with the answer:

```
bool a = NAND(x1, x2);
bool b = NAND(x2, x3);
bool y = NAND(a,b); return y;
```

Xiaoqiang wants to make sure that his solution is the shortest possible. Can you help him?

Input

The first line contains an integer T ($T \leq 20$) denoting the number of the test cases.

For each test case, there is one line containing 8 characters encoding the truth table of the function.

Output

For each test case, output a single line containing the minimum number of lines Xiaoqiang has to write.

Sample Input

```
1
00010011
```

Sample Output

```
4
```