

## 4845 Password

Shoulder-surfing is the behavior of intentionally and stealthily watching the screen of another person's electronic device, such as laptop computer or mobile phone. Since mobile devices prevail, it is getting serious to steal personal information by shoulder-surfing.

Suppose that we have a smart phone. If we touch the screen keyboard directly to enter the password, this is very vulnerable since a shoulder-surfer easily knows what we have typed. So it is desirable to conceal the input information to discourage shoulder-surfers around us. Let me explain one way to do this.

You are given a  $6 \times 5$  grid. Each column can be considered the visible part of a wheel. So you can easily rotate each column wheel independently to make password characters visible. In this problem, we assume that each wheel contains the 26 upper letters of English alphabet.

See the following Figure 1.

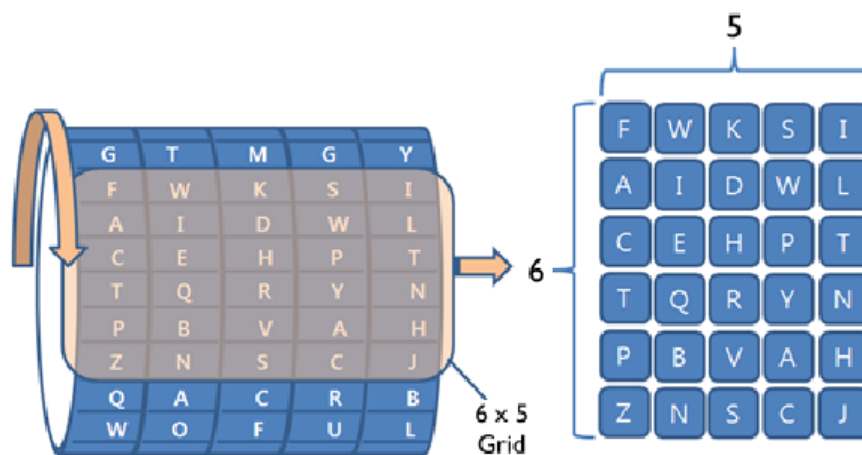


Figure 1.  $6 \times 5$  window clips a valid grid representation for a password.

Assume that we have a length-5 password such as  $p_1 p_2 p_3 p_4 p_5$ . In order to pass the authentication procedure, we should construct a configuration of grid space where each  $p_i$  appears in the  $i$ -th column of the grid. In that situation we say that the user password is accepted.

Let me start with one example. Suppose that our password was set 'COMPU'. If we construct the grid as shown in Figure 2 on next page, then the authentication is successfully processed.

In this password system, the position of each password character in each column is meaningless. If each of the 5 characters in  $p_1 p_2 p_3 p_4 p_5$  appears in the corresponding column, that can be considered the correct password. So there are many grid configurations allowing one password. Note that the sequence of letters on each wheel is randomly determined for each trial and for each column. In practice, the user is able to rotate each column and press "Enter" key, so a should-surfer cannot perceive the password by observing the  $6 \times 5$  grid since there are too many password candidates. In this  $6 \times 5$  grid space, maximally  $6^5 = 7,776$  cases are possible. This is the basic idea of the proposed password system against shoulder-surfers.

A	Y	G	S	U
D	O	M	R	A
C	P	F	A	S
X	B	O	D	G
W	D	Y	P	K
R	B	C	D	E

Figure 2. A valid grid representation for password 'COMPU'.

Unfortunately there is a problem. If a shoulder-surfer can observe more than two grid plate configurations for a person, then the shoulder-surfer can reduce the searching space and guess the correct password. Even though it is not easy to stealthily observe other's more than once, this is one weakness of implicit grid passwords.

Let me show one example with two observed configurations for a grid password. The user password is 'COMPU', but 'DPMAG' is also one candidate password derived from the following configuration.

A	Y	G	S	U
D	O	M	R	A
C	P	F	A	S
X	B	O	D	G
W	D	Y	P	K
P	R	X	W	O

C	B	O	P	T
D	O	S	B	G
G	T	R	A	R
A	P	M	M	S
W	S	X	N	U
E	F	G	H	I

Figure 3. Both of 'COMPU' and 'DPMAG' are feasible password .

You are given two configurations of grid password from a shoulder-surfer. Suppose that you have succeeded to stealthily record snapshots of the target person's device (e.g. smart phone). Then your next task is to reconstruct all possible passwords from these two snapshots. Since there are lots of password candidates, you are asked for the  $k$ -th password among all candidates in lexicographical order. In Figure 3, let us show the first 5 valid password. The first 5 valid passwords are 'ABGAG' , 'ABGAS', 'ABGAU', 'ABGPG' and 'ABGPS'.

The number  $k$  is given in each test case differently. If there does not exist a  $k$ -th password since  $k$  is larger than the number of all possible passwords, then you should print 'NO' in the output.

### Input

Your program is to read from standard input. The input consists of  $T$  test cases. The number of test cases  $T$  is given in the first line of the input. The first line of each test case contains one integer,  $K$ , the order of the password you should find. Note that  $1 \leq K \leq 7,777$ . Next the following 6 lines show the 6 rows of the first grid and another 6 lines represent the 6 rows of the second grid.

### Output

Your program is to write to standard output. Print exactly the  $k$ -th password (including 'NO') in one line for each test case.

The following shows sample input and output for three test cases.

### Sample Input

```
3
1
AYGSU
DOMRA
CPFAS
XBODG
WDYPK
PRXWO
CBOPT
```

DOSBG  
GTRAR  
APMMS  
WSXNU  
EFGHI  
5  
AYGSU  
DOMRA  
CPFAS  
XBODG  
WDYPK  
PRXWO  
CBOPT  
DOSBG  
GTRAR  
APMMS  
WSXNU  
EFGHI  
64  
FGHIJ  
EFGHI  
DEFGH  
CDEFG  
BCDEF  
ABCDE  
WBXDY  
UWYXZ  
XXZFG  
YYFYH  
EZWZI  
ZGHIJ

### Sample Output

ABGAG  
ABGPS  
NO