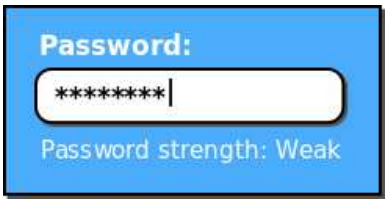
# Passwords




It's that time of the year again when you go back to work and need to choose new passwords for all your services. The rules enforced by the system administrators are very strict, and the password you choose must obey the following restrictions:

- It must contain only letters and digits.
- It must have between $A$ and $B$ characters (inclusive).
- It must have at least one lowercase letter, one uppercase letter and one digit.
- It cannot contain any word of a collection of forbidden words (the blacklist).

A word of the blacklist is considered to be contained in the password if it appears as a substring, that is, if it occurs as a consecutive sequence of characters (regardless of it being upper or lower case). For instance, `swerc` is a substring of `SwErC`, `2016swerc2016` or `SWERC16`, but it is not a substring of `ICPC` or `sw16erc`.

Additionally, for the purposes of avoiding the blacklist, you cannot use *l33t*. Specifically, some digits can be interpreted as letters, namely the 0 ('o'), 1 ('i'), 3 ('e'), 5 ('s') and 7 ('t'). This implies that for example `5w3rC` would be an occurrence of `swerc`, and that `abcL337def` contains the word `leet`.

You cannot stop thinking about all these rules and you wonder how many different valid passwords there are... Can you calculate how many passwords would obey these rules?

## Task

Given a blacklist with $N$ words and two integers $A$ and $B$, your task is to compute the number of different valid passwords that exist following the given constraints: made up of only letters and digits; length between $A$ and $B$ (inclusive); at least one lowercase letter, one uppercase letter, and one digit; no blacklisted substring. Since this number can be very big, compute it modulo 1 000 003.

## Input

The first line contains two integers, $A$ and $B$, specifying respectively the minimum and maximum length of the password. The second line contains an integer $N$, the number of words of the blacklist. The following $N$ lines each contains a string $W_i$ indicating a word in the blacklist. These words are formed only by lowercase letters.

## Constraints

| | |
|---|---|
| $3 \leq A \leq B \leq 20$ | Size of the password |
| $0 \leq N \leq 50$ | Number of words in the blacklist |
| $1 \leq \text{length}(W_i) \leq 20$ | Size of each blacklist word |

## Output

The output should contain a single line with an integer indicating the number of valid passwords modulo 1 000 003. A valid password is one that respects all of the given constraints.

## Sample Input

```
3 5
9
swerc
icpc
fbi
cia
bio
z
hi
no
yes
```

## Sample Output

```
607886
```

## Sample Explanation

In this case there are exactly 378 609 020 valid passwords and

$$378\,609\,020 \bmod 1\,000\,003 = 607\,886\,.$$

Some examples of valid passwords are: `aA1`, `B23tT`, `1g9K` or `B2j`.

Some examples of invalid passwords are: `aaA` (it does not contain digits), `12a` (it does not contain upper case letters), `a12A34` (length $> 5$) or `bB10` (contains `bio` as substring).