2025 T3 Launch Week Quokka Quontest Problems

UNSW CPMSoc

15 September 2025

Contents

Quokka Power [Maths]

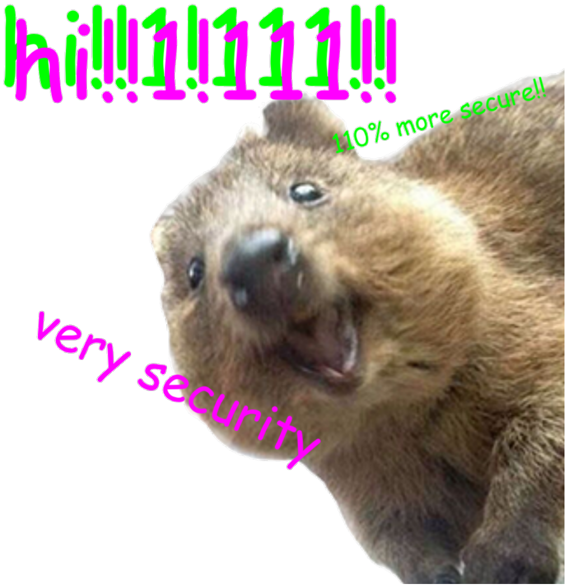
Basel the quokka loves odd numbers, and Rexpo loves even numbers.

Evaluate $1^6 \times 3^4 \times 5^2 \times 7^0$.

For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your best submission will be counted.

Quoccabank [Maths]

Larissa the quokka works at a bank, and to pass the time will roll some standard six sided dice. She only knows one game, which is to roll six dice at once and hope to get all the numbers from one to six exactly once. She asks you what the probability of winning her game is.



Submit your answer as an irreducible fraction, with the numerator and denominator separated by a slash. For example, if your answer is 0.8, you should submit "4/5".

For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your best submission will be counted.

Basic Instances [Maths]

Josh has been asked to paint all the letterboxes on a given segment of a street, and he's curious about the street numbers. He realises that the street numbers are always a consecutive segment of numbers, and poses some problems from there.

Subtask 1 (25% of points)

How many times does the digit 6 appear between 1 and 678?

Subtask 2 (25% of points)

How many numbers between 1 and 678 contain the digit 7?

Subtask 3 (25% of points)

How many times does the digit 0 appear between 6769 and 41420?

Subtask 4 (25% of points)

How many numbers between 15092025 and 21092025 contain the digit 9?

Submission

Submit your answer to the subtasks as a comma-separated list of integers. For example, if your answers to the subtasks are $1, 0, 2$ and 4, you should submit `1,0,2,4`. Note that if you have not solved a subtask, you can submit a dummy answer for that subtask. For example, if your answer to the first subtask is 1, you could submit `1,0,0,0`.

F(Quokka) [Maths]

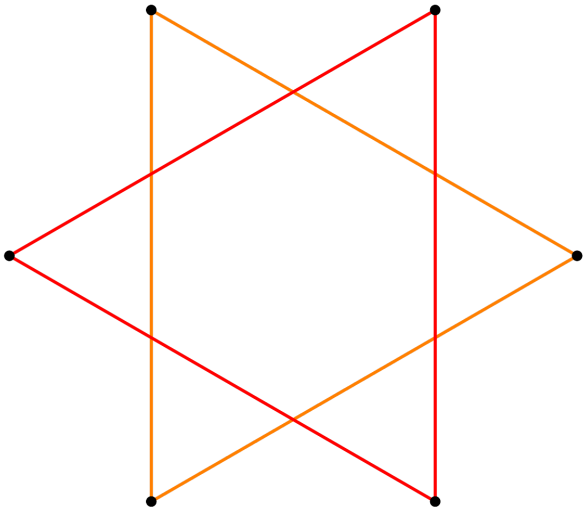
Let $f(x)$ represent the sum of digits x. What is the smallest positive integer x such that $(f(x)+1)^2$ is a factor of x?

For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your best submission will be counted.

Quokkstar [Maths]

Rob the quokka is playing with an infinite number of unit hexagrams, which can be described as a hexagon of side length 1 with a triangle of side length 1 affixed to each of its sides. He is trying to cover various squares of side length x. Let $f(x)$ be the maximum proportion of area of the square that be covered with non-overlapping unit hexagrams.

Evaluate $\lim_{x \to \infty} f(x)$.



Submit your answer as an irreducible fraction, with the numerator and denominator separated by a slash. For example, if your answer is 0.8, you should submit “4/5”.

For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your best submission will be counted.

Bizarre Pea Stall [Maths]

Simon is playing some carnival games, and one particularly odd stall requires him to toss a pea chip into a bottlecap. He is successful 40% of the time, and misses the other 60% of the time. His score starts at zero. Every time he manages to put the chip into the cap, his score goes up by one. Every time he misses, his score decreases by up to two, but it will never go negative.

What is the expected number of tosses that Simon will need to reach a score of 4?

Submit your answer as an irreducible fraction, with the numerator and denominator separated by a slash. For example, if your answer is 0.8, you should submit “4/5”.

For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your best submission will be counted.

Triples [Maths]

A cute quokka joined CPMSoc this year and while being part of CPMSoc, he learned that some triples are very interesting. More specifically, the set of triples (x_i, y_i, z_i) that have the following properties:

- $x_i + y_i + z_i = 20$ for all i,
- $x_i \neq x_j$ for any i, j where $i \neq j$,
- $y_i \neq y_j$ for any i, j where $i \neq j$,
- $z_i \neq z_j$ for any i, j where $i \neq j$,
- And lastly, the numbers x_i, y_i, z_i are all non-negative integers for all i, $(x_i, y_i, z_i \in \mathbb{Z}^+ \cup \{0\})$.

To satiate his curiosity, he wants you to find the largest set of triples that satisfy the above properties. You can provide the $3k$ numbers separated by either an empty-space or a comma. Your numbers will be of the following form:

```
x_1, y_1, z_1
x_2, y_2, z_2
...
x_k, y_k, z_k
```

For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your best submission will be counted.

Fair and Square [Maths]

Andy and Brian are playing a game on a square grid. They take turns placing their initial ('A' or 'B') into an empty cell on the grid, and win if they are the first to claim all four corners of any square larger than the trivial 1 x 1. Note that this square must be axis aligned (for instance, $(0, 1), (1, 0), (1, 2), (2, 1)$ does not form a winning square).

This game can be played either on a 36 by 36 board, or an infinitely large board. For both versions, either find a winning strategy for one player, or prove that the game will end in a draw after optimal play.

Up to 7 marks will be awarded for proving the correct result of one version, and up to 10 marks will be awarded for proving the correct result in both versions.

For all the tasks, including this one, you can submit as many times as you like with no penalty! Only your best submission will be counted.

Quokka Pairs [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

N quokkas are performing a dance together!

Each quokka has been given a score, A_i based on how well they danced!

At the end of the dance, judges are handing out a special prize, which belongs to a pair of Quokkas whose scores add up to a certain score K.

Your job is to see if this pair exists! Output 'Y' if there exists a pair that can receive prizes, otherwise output 'N'.

Input

- The first line of input contains two integers N, representing the number of quokka dancers, and K, representing the score that a pair of Quokkas needs to add up to to win the special prize.
- The second line contains N integers, $A_1, \ldots, A_N$, represeting the scores of each quokka.

Constraints

For all test cases:

- $1 \leq N \leq 1000$,
- $0 \leq K \leq 10^9$,
- $0 \leq A_i \leq 10^9$ for $1 \leq i \leq N$.

Additionally:

- For Subtask 1 (50% of points): K is guaranteed odd.
- For Subtask 2 (50% of points): there are no additional constraints.

Output

- Output 'Y' if there exists a pair that can receive prizes, otherwise output 'N'.

Templates

You should read from standard input and write to standard output.

In Python, you could use the following code.

```
# Taking inputs, already done! :D
N, K = map(int, input().split())
A = list(map(int, input().split()))

flag = False
# Write your code here


# Printing output
if flag:
    print("Y")
else:
    print("N")
```

In C or C++, you could use the following code.

```
// Taking inputs, already done! :D
int N, K; scanf("%d%d", &N, &K);
int A[N]; for (int i = 0; i < N; i++) scanf("%d", &A[i]);

bool flag;
// Write your code here


// Printing output
if (flag == true) {
    printf("Y\n");
} else {
    printf("N\n");
}
```

Sample Input 1

```
7 67
41 6 7 9 26 4 1
```

Sample Output 1

```
Y
```

Explanation 1

In this case, K is 67, so quokka dancers with the scores 41 and 26 can add up to K, winning their special prize.

Scoring

For each subtask (worth 50% and 50% of points, as per the Constraints section), your program will be run on multiple secret test cases one after another, and if it produces the correct output for **all** test cases, it solves that subtask. Your program will receive the points for each subtask it solves. Recall that your final score on the task is the score of your highest scoring submission.

Alphabet [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

You have just landed in Quokka kingdom and Queen Quokka has given you a task - to sort a string in alphabetical order.

However, you can't sort the string in *our* alphabetical order, you have to sort it in *their* alphabetical order. You are given the order of their alphabet as input.

Input

- The first line contains a string representing the alphabet order used in Quokka kingdom,
- The second line contains a string S that you need to sort according to the given alphabet order.

Constraints

For all test cases:

- The alphabet string contains exactly 26 lowercase English letters, each appearing exactly once
- $1 \leq |S| \leq 1000000$ where $|S|$ is the length of string S
- String S contains only lowercase English letters

Output

- Output the string S sorted according to the given alphabet order.

Templates

You should read from standard input and write to standard output.

In Python, you could use the following code.

```python
# Taking inputs, already done! :D
alphabet = input().strip()
S = input().strip()

result = ""
# Write your code here


# Printing output
print(result)
```

In C or C++, you could use the following code.

```cpp
// Taking inputs, already done! :D
char alphabet[27]; scanf("%s", alphabet);
char S[1000001]; scanf("%s", S);

string result;
// Write your code here


// Printing output
printf("%s\n", result.c_str());
```

Sample Input 1

```
zyxwvutsrqponmlkjihgfedcba
hello
```

Sample Output 1

```
ollhe
```

Explanation 1

The alphabet order is reversed (z comes first, a comes last). So when we sort “hello” according to this order, ‘o’ comes first (position 11), followed by ‘l’ (position 14), ‘l’ (position 14), ‘h’ (position 18), and ‘e’ (position 21). The sorted result is “ollhe”.

Sample Input 2

```
abcdefghijklmnopqrstuvwxyz
world
```

Sample Output 2

```
dlorw
```

Explanation 2

The alphabet order is the standard English alphabet (a comes first, z comes last). When we sort “world” according to this order, ‘d’ comes first, followed by ‘l’, ‘o’, ‘r’, and ‘w’. The sorted result is “dlorw”.

Sample Input 3

```
qwertyuiopasdfghjklzxcvbnm
programming
```

Sample Output 3

```
rriopaggnmm
```

Explanation 3

The alphabet order is based on the QWERTY keyboard layout. When we sort “programming” according to this order, the letters are arranged as: r, r, i, o, p, a, g, g, n, m, m. The sorted result is “rriopaggnmm”.

Scoring

Your program will be run on both sample cases and 20 secret cases one after another, and if it produces the correct output for **all** test cases, it solves this task. Recall that your final score on the task is the score of your highest scoring submission.

UNSWap [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

Isaiah is computing the sum of an array a, which contains n integers: $a_1, a_2, \ldots, a_n$.

After finding the sum, he can perform exactly one digit swap, which is done by swapping around any two digits. Digit sums can either be within the same number, or across different numbers. The only restriction is numbers cannot start with the digit 0, unless the number is a single digit 0.

How many different digit swaps are there which do not change the sum of array A? Since the answer can get big, please give your answer modulo $1,000,000,007$.

Input

- The first line of input contains an integer n, which represents the number of
- The second line contains N space separated integers, representing A.

Constraints

For all test cases:

- $1 \leq n \leq 10^5$,
- $0 \leq a_i \leq 10^9$ for $1 \leq i \leq n$.

Additionally:

- For Subtask 1 (50% of points): $n \leq 100$,
- For Subtask 2 (50% of points): there are no additional constraints.

Output

- Output a single integer, which represents the number of digit swaps which do not change the sum of array A.

Templates

You should read from standard input and write to standard output.

In Python, you could use the following code.

```
# Taking inputs, already done! :D
n = int(input())
A = list(map(int, input().split()))

ans = 0
# Write your code here


# Printing output
print(ans)
```

In C or C++, you could use the following code.

```
// Taking inputs, already done! :D
int n; scanf("%d", &n);
int A[n]; for (int i = 0; i < n; i++) scanf("%d", &A[i]);

int ans;
```

```
// Write your code here


// Printing output
printf("%d\n", answer);
```

Sample Input 1

```
2
123 330
```

Sample Output 1

```
6
```

Explanation 1

These are four pairs of numbers with a sum of 453: $323 + 130, 133 + 320, 120 + 333, 123 + 330$. Importantly, there are three swaps that result in $123 + 330$. Therefore, there are a total of 6 valid digit swaps.

Sample Input 2

```
1
110
```

Sample Output 2

```
1
```

Explanation 2

The only valid digit swap is between the 1 in the hundreds position and the 1 in the tens positions.

Scoring

For each subtask (worth 50% and 50% of points, as per the Constraints section), your program will be run on multiple secret test cases one after another, and if it produces the correct output for **all** test cases, it solves that subtask. Your program will receive the points for each subtask it solves. Recall that your final score on the task is the score of your highest scoring submission.

Quokkas and Koalas [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

At Quokkas and Koalas Love Society, your members are mingling based on how much they like Quokkas and Koalas.

Each member has a value Q_i which indicates how much they like Quokkas and a value K_i which indicates how much they like Koalas. Two members i and j have a Quokka love coefficient of $Q_i \cdot Q_j$ and a Koala love coefficient of $K_i \cdot K_j$. Two members are friends if these coefficients are equal.

Output how many pairs of friends there are.

Input

- The first line of input contains an integer N, representing the number of members.
- The second line contains N space separated integers, representing $Q_1, Q_2, \ldots, Q_N$.
- The third line contains N space separated integers, representing $K_1, K_2, \ldots, K_N$.

Constraints

For all test cases:

- $1 \leq N \leq 100,000$,
- $0 \leq Q_i, K_i \leq 1,000,000,000$ for $1 \leq i \leq N$.

Output

- Output a single integer, which represents the number of pairs of friends.

Templates

You should read from standard input and write to standard output.

In Python, you could use the following code.

```python
# Taking inputs, already done! :D
N = int(input())
Q = list(map(int, input().split()))
K = list(map(int, input().split()))

ans = 0
# Write your code here


# Printing output
print(ans)
```

In C or C++, you could use the following code.

```c
// Taking inputs, already done! :D
int N; scanf("%d", &N);
long long Q[N]; for (int i = 0; i < N; i++) scanf("%lld", &Q[i]);
long long K[N]; for (int i = 0; i < N; i++) scanf("%lld", &K[i]);

long long ans;
// Write your code here
// Printing output
printf("%lld\n", ans);
```

Sample Input 1

```
6
1 2 2 5 5 6
1 1 4 0 5 4
```

Sample Output 1

```
2
```

Explanation 1

We need to find pairs (i, j) where $Q_i \cdot Q_j = K_i \cdot K_j$.

For members 1 and 5, $Q_1 \cdot Q_5 = 1 \cdot 5 = 5$ and $K_1 \cdot K_5 = 1 \cdot 5 = 5$ so they are friends. For members 2 and 3, $Q_2 \cdot Q_3 = 2 \cdot 2 = 4$ and $K_2 \cdot K_3 = 1 \cdot 4 = 4$ so they are also friends. These are our only pairs of friends, so the answer is 2.

Sample Input 2

```
3
2 3 0
3 2 0
```

Sample Output 2

```
3
```

Explanation 2

All three pairs are friends, so the answer is 3.

Scoring

Your program will be run on both sample cases and 7 secret cases one after another, and if it produces the correct output for **all** test cases, it solves this task. Recall that your final score on the task is the score of your highest scoring submission.

CPM Banner [Programming]

Program time limit: 1 second

Program memory limit: 512 MB

You are designing a banner for CPM (Competitive Programming Masters) and want to make it as short as possible while still being recognizable. The banner should contain the letters "CPM" as a subsequence exactly N times.

You are given an initial string that must be a prefix of your final banner. Given N and the initial string, output the shortest string where "CPM" appears as a subsequence **exactly** N times and starts with the given initial string. If there are multiple strings of equal length, output the lexicographically smallest one. If it's impossible to satisfy the constraints, output "impossible".

A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. For example, "CPM" appears as a subsequence in "CCPMM" four times (after removing elements at positions $(1, 4), (1, 5), (2, 4) or (2, 5)$).

Input

- The first line of input contains an integer N, representing the number of times "CPM" should appear as a subsequence.
- The second line contains an integer L, representing the length of the initial string.
- The third line contains a string S of length L (consisting only of characters 'C', 'P', 'M'), representing the initial string that must be a prefix of the answer.

Constraints

For all test cases:

- $1 \leq N \leq 200$
- $1 \leq L \leq 500$ where L is the length of string S
- String S contains only characters 'C', 'P', 'M'

Output

Output a single string, which is the shortest string containing "CPM" as a subsequence exactly N times and starting with the given initial string. If there are multiple strings of equal length, output the lexicographically smallest one. If it's impossible to satisfy the constraints, output "impossible".

Templates

You should read from standard input and write to standard output.

In Python, you could use the following code.

```
# Taking inputs, already done! :D
N = int(input())
L = int(input())
S = input().strip()

result = ""
# Write your code here


# Printing output
print(result)
```

In C or C++, you could use the following code.

```
// Taking inputs, already done! :D
int N; scanf("%d", &N);
int L; scanf("%d", &L);
char S[505]; scanf("%s", S);

string result;
// Write your code here


// Printing output
printf("%s\n", result.c_str());
```

Sample Input 1

```
1
3
CPM
```

Sample Output 1

```
CPM
```

Explanation 1

We have an initial string “CPM” and need exactly 1 occurrence of “CPM” as a subsequence. The shortest string is “CPM” itself, which has length 3 and starts with the given initial string.

Sample Input 2

```
2
2
CC
```

Sample Output 2

```
CCPM
```

Explanation 2

We have an initial string “CC” and need exactly 2 occurrences of “CPM” as a subsequence. The shortest string satisfying this is “CCPM”, where “CPM” appears at positions (1, 3, 4) and (2, 3, 4).

Sample Input 3

```
1
3
CPP
```

Sample Output 3

```
impossible
```

Explanation 3

We have an initial string “CPP” and need exactly one occurance of “CPM”. This is impossible as we must add an “M” at some point, but adding one would create at least two “CPM”s.

Scoring

Your program will be run on both sample cases and 20 secret cases one after another, and if it produces the correct output for **all** test cases, it solves this task. Recall that your final score on the task is the score of your highest scoring submission.

Magic [Programming]

Time limit: 1 second | Memory limit: 512 megabytes

Magical quokkas keep changing the color of their fur in mysterious ways. At first, every quokka starts with the plain color (color 0). Then, at each step, the quokka may change into another color. Eventually, the quokka will settle into a final color, which never changes again.

Some colors are transitional: from such a color, the quokka's color might transform into one of several next colors, with fixed probabilities. (a color may change it's color into itself. in other words, there can be i = j transition) The colors are called final if it's one of the possible last color.

You are given the list of transitions between colors: for each observed transition, you know how many times a quokka changed from one color into another.

Formally: - If a line says "i j c", it means that from color i, the quokka was seen changing into color j exactly c times. - The probability of going from color i to color j is then proportional to c. - If a color does not change into a different color, then it is a final color.

It is guaranteed that from every color there is some sequence of changes leading to a final color.

Your task is to determine the exact probability that a magical quokka starting in color 0 will eventually end up in each final color.

The answer should be given as a list of integers: the numerators of the probabilities for each final color. Fractions must be in simplest form.

Constraints and Scoring

For all test cases:

- $1 \leq N \leq 10$
- $1 \leq M \leq N^2$
- $0 \leq i, j < N$
- $1 \leq c \leq 50$ The test cases are divided into subtasks. Your program must output a correct answer for **every test case** in a subtask to receive the points for that subtask.
- For Subtask 1 (20% of points): There is no cyclic transition.
- For Subtask 2 (80% of points): There are no additional constraints

Input

- The first line of input contains two integers N and M.
- The following M lines contain three integers i, j, and c indicating that there is a transition between i and j with weight c.

Output

Output the probabilities for each final color in increasing order. The probability must be in it's most simple form. If the answer is 1 or 0, it should be printed without any denominator.

If you are using C++, since the answer may be large, you should use the `__int128` data type.

Sample Input 1

```
4 3
0 1 1
0 2 1
```

```
0 3 1
```

Sample Output 1

```
1/3 1/3 1/3
```

Sample Input 2

```
4 5
0 1 1
0 2 9
0 3 8
1 2 4
1 3 5
```

Sample Output 2

```
85/162 77/162
```

Word Count [Programming]

Program time limit: 2 seconds

Program memory limit: 512 MB

Joceline has a grid of size $N \cdot M$ filled with letters of the alphabet. Each row except for the first can be left-shifted by some value X - that is, replace the i^{th} value with the $(i + X)\%M^{\text{th}}$ value.

You are given a dictionary of D words, each N long. The wordcount of the grid is the number of columns with valid words.

You are given Q queries of the form:

- You are given values i and j, you then shift the i^{th} row by j
- Joceline shifts every other row except for the first by a random value
- Output the expected wordcount after these shifts

The grid is reset after each query.

Input

- The first line contains two integers N and M, representing the dimensions of the grid.
- The next N lines each contain a string of length M consisting of lowercase characters, representing the grid.
- The next line contains an integer D, representing the number of words in the dictionary.
- The next D lines each contain a string of length N consisting of lowercase characters, representing the dictionary words.
- The next line contains an integer Q, representing the number of queries.
- The next Q lines each contain two integers i and j, representing the row to shift and the shift amount.

Constraints

For all test cases:

- $1 \leq N \leq 50$
- $1 \leq M, D \leq 10'000$
- $1 \leq Q \leq 500'000$
- $1 \leq i \leq N$
- $0 \leq j < M$
- All strings contain only lowercase letters
- All words in the dictionary are distinct

Scoring

- For Subtask 1 (10% of points): $1 \leq M \leq 500$
- For Subtask 2 (10% of points): $1 \leq M \leq 1000$
- For Subtask 3 (10% of points): $1 \leq M \leq 5000$
- For Subtask 4 (70% of points): There are no additional constraints

Output

For each query, output a single line containing the expected wordcount after the shifts as a fraction modulo 998244353. Specifically, if the expected wordcount is $\frac{P}{Q}$ where P and Q are coprime integers, output $(P \cdot Q^{-1})$ mod 998244353 where Q^{-1} is the modular inverse of Q modulo 998244353.

Sample Input 1

```
3 4
abcd
baca
bcdb
6
abb
acb
adb
cbb
aab
bcb
2
2 1
3 2
```

Sample Output 1

```
1
499122177
```

Explanation 1

Query 1 (Row 2 is shifted left by 1)

- **Column 1:**
  - Row 1, Col 1 = `a`.
  - After shifting Row 2 by 1, Row 2, Col 1 = `a`.
  - Required pattern is `aa?` -> the only dictionary word that fits is `aab`.
  - Row 3 is random; probability it is `b`: $2/4 = 1/2$.
- **Column 2:**
  - Row 1, Col 2 = `b`.
  - After shifting Row 2 by 1, Row 2, Col 2 = `c`.
  - Required pattern is `bc?` -> the only dictionary word that fits is `bcb`.
  - Row 3 is random; probability it is `b`: $2/4 = 1/2$.
- **Column 3:**
  - Row 1, Col 3 = `c`.
  - After shifting Row 2 by 1, Row 2, Col 3 = `a`.
  - Required pattern is `ca?` -> no dictionary word matches. Probability: 0.
- **Column 4:**
  - Row 1, Col 4 = `d`.
  - After shifting Row 2 by 1, Row 2, Col 4 = `b`.
  - Required pattern is `db?` -> no dictionary word matches. Probability: 0.

Sum of probabilities: $1/2 + 1/2 + 0 + 0 = 1$. Answer: 1.

Query 2 (Row 3 is shifted left by 2)

- **Column 1:**
  - Row 1, Col 1 = `a`.
  - After shifting Row 3 by 2, Row 3, Col 1 = `d`.
  - Required pattern is `a?d` -> no dictionary word matches. Probability: 0.
- **Column 2:**
  - Row 1, Col 2 = `b`.

  - After shifting Row 3 by 2, Row 3, Col 2 = `b`.
  - Required pattern is `b?b` -> the only dictionary word that fits is `bcb`.
  - Row 2 is random; probability it is `c`: 1/4.
- **Column 3:**
  - Row 1, Col 3 = `c`.
  - After shifting Row 3 by 2, Row 3, Col 3 = `b`.
  - Required pattern is `c?b` → the only dictionary word that fits is `cbb`.
  - Row 2 is random; probability it is `b`: 1/4.
- **Column 4:**
  - Row 1, Col 4 = `d`.
  - After shifting Row 3 by 2, Row 3, Col 4 = `c`.
  - Required pattern is `d?c` -> no dictionary word matches. Probability: 0.

Sum of probabilities: $1/4 + 1/4 + 0 + 0 = 1/2$. Answer: 499122177.