DiverseFlowers (250)

There are N flowers in a garden. Each flower has two properties: size (big or small) and color (red, green or blue). We will mark the type of a flower with one of six letters: R, G, B, r, g, b. For example, 'g' denotes a small green flower and 'B' denotes a big blue flower.

You will pick some flowers from the garden and arrange them into flower bouquets, each containing exactly **K** flowers. A bouquet must contain at least one flower of each color. A bouquet must also contain at least one flower of each size. And obviously, each individual flower can only be used in at most one bouquet.

Can you maximize the number of created bouquets?

You are given a String **S** with N characters that describe types of flowers in the garden. Return a String[] where each element represents one bouquet as a String with exactly **K** characters. If multiple solutions maximize the number of bouquets, return any of them. The order of bouquets doesn't matter, and neither does the order of flowers in a bouquet.

Constraints

- **S** will contain exactly N characters.
- N will be between 3 and 5000, inclusive.
- **K** will be between 3 and N, inclusive.
- Each character in **S** will be one of six: R, G, B, r, g, b.

Example

“BBgbrGBBB” 3
Returns: {"Bgr" }

BagsOfNumbers (500)

Alice and Bob have *n* bags, each containing some, possibly zero, positive integers: ones, twos, threes, and fours. They play a game using these bags. Alice goes first, and then they alternate turns. In one turn, a player first selects exactly one of the bags. Then she or he removes some non-empty subset of numbers whose sum is divisible by four. Whoever is not able to make a move, loses.

The number of ones, twos, threes and fours in bag number i is **a**[i], **b**[i], **c**[i] and **d**[i], respectively. For each of the 2^n - 1 non-empty subsets of bags, consider the game played with only these bags. Determine how many of these games are won by Alice if they both play optimally.

Constraints

- *n* will be between 1 and 50, inclusive.
- The arrays **a**, **b**, **c** and **d** will have exactly *n* elements each.
- Each element of each array will be an integer between 0 and 10^{18}, inclusive.

Example

2
{13,13}
{12,12}
{25,25}
{9,9}

Returns: 2

Explanation:
When there is only single bag, Alice can win by removing all integers in it (their sum is 148, which is divisible by four). When playing with both bags, Bob can simply mirror Alice's moves to win.

FourDistinctDigits (1000)

You are given positive integers **N**. **D** and **B**. Find any positive integer X with the following properties:

- X is a multiple of **N**.
- When written in base **B**, X has exactly **D** digits.
- When written in base **B**, X has at most *four distinct* digits.

Return a String of length **D** containing the base-**B** representation of X. Use 'A'-'Z' for digits 10-35, and 'a'-'z' for digits 36-61. For the given constraints a solution always exists, and any valid solution will be accepted.

Constraints

- **N** will be between 1 and 10^11, inclusive.
- **D** will be between 40 and 120, inclusive.
- **B** will be between 5 and 62, inclusive.

Example

4247
40
10

Returns: "4247424742474247424742474247424742474247"