Rules and Regulations.

This competition is being facilitated with the use of HackerRank to submit and grade solutions. The following languages will be accepted on HackerRank: C++, Java, Python3, Python2, and C. We guarantee that all problems are solvable within the time limits.

The use of official language documentation and manuals is permitted, along with general tutorial sites. However, question-and-answer sites and forums are NOT PERMITTED. Sites like StackExchange, StackOverflow, and others will be blocked on the competition network; if caught, any attempt to evade this block will result in immediate disqualification.

Additionally, generative AI is NOT allowed under any circumstances. Tools such as ChatGPT and Github Copilot are not permitted. If caught, any attempt to use these tools will result in immediate disqualification.

Furthermore, teams may not accept assistance or advice from anyone not authorized to provide such assistance, including, but not limited to, other teams and chaperones.

Teams will be given example I/O test cases, but the I/O test cases used for grading will be hidden. The correctness of a solution is based on whether all test cases are passed; there is no partial credit. The examples provided will clearly indicate the expected output format.

Grading is done automatically; any deviation from the expected output will result in an incorrect result. This includes any extra whitespace, characters, or return/newline characters in the given output.

Should a tie impact the results of this contest, the time taken will be used as a tie-breaker. Thus, teams should keep time management in mind.

There are eight questions that are posed in the following pages. They progressively get more difficult; the expected difficulty of each question is given to help you choose how to allocate your time.

One laptop per team.

Question 1: Tip the Bill

You are eating at the famous Prowling Tiger Restaurant with a friend and plan to cover the total cost of both your meals. You also have to account for a percentage tip which is added to the total bill.

Given prices for two meals, *a* and *b*, and *n*, the percent tip, find the total cost of the bill

The first line contains *a*, representing the cost of your meal. The second line contains *b*, representing the cost of your friend's meal. The third line contains *n*, which is the percent tip added to the total bill. Output the total cost of the two meals with tip included. The number *n* will be inputted without a percent sign. We will guarantee that *a, b* < 200, 0<= *n* <=100, and that all inputs and outputs are integers.

Example 1:
Input:
40
25
20
Output:
78

Question 2: Decipher the Recipe

Jim the Sous Chef is busy cooking a meal, and is given a string of length *n*. Each character in the string corresponds to an ingredient in the recipe he is following, so it's crucial he sorts the string. Each string contains only lowercase letters a-z, where each letter represents a given ingredient in the recipe that Jim the Sous Chef needs. Jim must sort these ingredients in descending order, based on the frequency of the ingredient. The frequency of a character is the number of times it appears in the string. If two characters have the same frequency, sort them alphabetically.

We guarantee that all of the strings contain only lowercase letters of the English alphabet and $n < 1000$.

The first line of input is the integer *n*, and the second is a lowercase string of length *n*.

Example 1:
Input:
12
jimmybillbob
Output:
bbbiillmmjoy

Question 3: Saucy Experimentation

Pablo the Saucier (a chef in charge of making sauces) wants to experiment how sauces will reduce when they simmer. The sauce will reduce by the amount, in milliliters, given by the following formula:

$((T^2)/100.0) * 0.01 * C * 18.0$

where T represents the temperature of the burner, and C is a constant that represents the initial viscosity of the particular sauce. All sauces simmer for the same amount of time, reflected by constants in the expression.

The first line of input will contain C. The second line of input will contain n. The third line of input will contain n temperatures T, which are all three-digit integers. It is guaranteed that $n < 100$. For each temperature, find the amount the sauce reduces by, and then output the sum of all the results.

Example 1:
Input:
12.5
4
120,135,240,300
Output: 4056
Explanation:

$(120^2)/100 * 0.01 * 12.5 * 18 = 324$
$(135^2)/100 * 0.01 * 12.5 * 18 = 410.0625$
$(240^2)/100 * 0.01 * 12.5 * 18 = 1296$
$(300^2)/100 * 0.01 * 12.5 * 18 = 2025$

$324 + 410.0625 + 1296 + 2025 = 4055.0625 \sim= 4056$

Question 4: The Maître D's Revenge

Maximus the Maître D' was recently fired for fraternizing with customers!!! Maximus decided to get his revenge by locking the chef's freezer/fridge combo and taking the key. The freezer is arranged as a 9 by 9 square array of individual cabinets, and each cabinet has an ingredient it is labeled with. To help us solve the puzzle, Maximus left us a list of positive and negative integers, and a list of positive non-zero integers. For each number in the first list with an even index in the list (assuming 0 is even), you must move on the horizontal axis. Additionally, a negative number means you move to the left and conversely a positive number means to the right. For each number with an odd index in the list, you must move on the vertical axis. Additionally, a negative number means you must move downwards and conversely a positive number means upwards. To appease Maximus' sadistic attitude, you must tell him the ingredients you will land on at intervals determined by the second list. For each number *c* in the second list, you must read out the ingredient you're located on after you execute *c* instructions from the last instruction executed (the first *c*-length group of instructions will start with the first instruction). Note that your position does not reset after each interval, and that you start at the top left corner of the array. Furthermore, you must wrap around to the other side of an array if an instruction takes you past the boundary.

The first line of input is a list of *d* numbers, separated by spaces, where $0 > d > 10^3$ and where each number *n* is $-10^9 < n < 10^9$. The number *n* represents how far you move, and the index of *n* in the list decides the direction you move in as explained above. If *n* exceeds the length of the array, the pointer should wrap around to the other side. The next line of input contains a list of *m* total positive integers, each separated by spaces, where $0 < m <= d$. We guarantee that the sum of the intervals won't exceed how many instructions there are. The next 9 lines are lists of 9 ingredients separated by semicolons, representing the 9 by 9 array of ingredient cabinets. Output the ingredients found on the intervals on their own individual lines.

Example on next page:

Example 1:
Raw Input:
2 -3 4 5 20
3 2
Apple; Banana; Tomato; Bread; Cheese; Cheddar; Churro; Flower; Egg
Pasta; Celery; Parsley; Basil; Chicken; Beef; Steak; Mozzarella; Thime
Chocolate; Flour; Gluten; Baking Soda; Vinegar; Olive Oil; Butter; Cream; Baking Powder
Ham; Apple Seeds; Bitter Almonds; Pumpkin Pie; Apple Pie; Pecan Pie; Prosciutto; Sausage; Egg Yolk
Egg White; Pesto; Tofu; Goat; Strawberry; Herring; Salmon; Fish; Rice
Sugar; Simple Syrup; High-fructose corn syrup; Cardamom; Peanut; Cashew; Pecan; Walnut; Chestnuts
Acorn; Watermelon; Allspice; Nutmeg; Oregano; Cinnamon; Avocado; Avocado Oil; Elderberries
Yeast; Apple Jacks; Cinnamon Toast Crunch; Red Cabbage ; Cilantro; Jalapeno; Turkey; Skinned Chicken; Mutton Liver
Chopped Liver; Crab; Chicken Stock; Chicken Liver; Chicken Entrails ; Quail; Mutton; Surstromming; Shark

Output:
Prosciutto
Mutton Liver

Line 1 (interpreted):
2 right
3 down
4 right
5 up
20 right
Line 2 (interpreted):
Output the ingredients you’re located on after executing instructions 1-3 (2 right, 3 down, 4 right) and after executing instructions 4-5 (5 up, 20 right) as a list.

(Array stylized)

Apple	Banana	Tomato	Bread	Cheese	Cheddar	Churro	Flower	Egg
Pasta	Celery	Parsley	Basil	Chicken	Beef	Steak	Mozzarella	Thime
Chocolate	Flour	Gluten	Baking Soda	Vinegar	Olive Oil	Butter	Cream	Baking Powder
Cherry Pits	Apple Seeds	Bitter Almonds	Pumpkin Pie	Apple Pie	Pecan Pie	**Prosciutto**	Sausage	Egg Yolk
Egg White	Pesto	Tofu	Goat	Strawberry	Herring	Salmon	Fish	Rice
Sugar	Simple Syrup	High-fructose corn syrup	Cardamom	Peanut	Cashew	Pecan	Walnut	Chestnuts
Acorn	Watermelon	Allspice	Nutmeg	Oregano	Cinnamon	Avocado	Avocado Oil	Elderberries
Yeast	Apple Jacks	Cinnamon Toast Crunch	Red Cabbage	Cilantro	Jalapeno	Turkey	Skinned Chicken	**Mutton Liver**
Chopped Liver	Crab	Chicken Stock	Chicken Liver	Chicken Entrails	Quail	Mutton	Surstromming	Shark

Question 5: Back to “Base”ics.

After Maximus the Maître D’ was fired, head chef Gourmand decided it would be better to use an electronic ordering system, where each number corresponds to a dish. But, the electronic ordering system has gone haywire! Gourmand suspects that Maximus was still salty about his dismissal and sabotaged the ordering system out of revenge.

The broken system returns an n-digit number in base b. You must convert this base-b number to a number in base k. The sum of the base-10 interpretations of the digits of the base-k number is the original number in the electronic ordering system.

The first line of input will contain b, which represents the base of the starting number. The second line will contain k, which represents the base of the number you need to convert to. The third line of input will contain an n-digit number, which is the starting number in base b.
We guarantee that $2 \leq b \leq 16$, $2 \leq k \leq 16$, $n < 600$. Digits for bases greater than 10 are as follows A=10, B=11, C=12, D=13, E=14, F=15.

Example 1:
Input:
3
2
1122
Output:
3
Explanation:
$1122_3 = 44_{10} = 101100_2$
$1_{10} + 0_{10} + 1_{10} + 1_{10} + 0_{10} + 0_{10} = 3_{10}$

Example 2:
Input: 12
15
21AB34
Output:
26
Explanation:
$21AB34_{12} = 537304_{10} = A9304_{15}$
$10_{10} + 9_{10} + 3_{10} + 0_{10} + 4_{10} = 26$

Question 6: Fibonacci Flambe

After looking through the pantry, Head Chef Gourmand found a piece of paper with cryptic numbers on it. Each piece of paper has the numbers on the top, *a, b,* and *k* on different lines, the last line has *k* integers separated by spaces. *a* and *b* represent the first two digits of a Fibonacci sequence where a is the first digit and b is the second. The third digit in the sequence (f_3) would be a + b and the fourth would be b + f_3. Each number in the sequence is a sum of the previous two numbers

Below is an example of the first 10 digits of the sequence where a = 1 and b = 3

1, 3, 4, 7, 11, 18, 29, 47, 76, 123…

Each of the *k* integers, $n_1,\ldots,n_k$, in the last line represents a term in a Fibonacci sequence. To decode the message, for each *n* in $n_1, \ldots, n_k$, find the last two digits of the *n*th term of the given Fibonacci sequence. Once those digits are found, convert them into their decoded character using ASCII values (see chart below). Return the message found by decoding all of the indexed numbers in the Fibonacci sequence.

The first line of input will contain *a*, which is the first number of the Fibonacci sequence. The second line will contain *b*, which is the second number of the sequence. The third line of input will contain *k*, which is the number of terms you will need to decode. The fourth line of input will contain *k* numbers, which are the terms of the sequence you need to calculate and then decipher.

It is guaranteed that *a* and *b* will be less than 200 and that *k* will be less than 10^4. It is also guaranteed each *n* will be less than 1000.

Input Format:
a
b
k
n_1 n_2 … n_{k-1} n_k

Example 1:

Input:

2
3
11
52 29 46 46 56 172 14 56 121 46 22 11

Output: HELLO WORLD!

Explanation:

Using f to represent the sequence starting with 2 and 3 (a and b) we can start decoding with 52

The first step is finding the 52nd term in the sequence
$f_1 = 2$, $f_2 = 3$, $f_3 = 5$ … f_{52}=86267571272

Once that term is found the last two digits(72) are converted to their character representation using the standard ASCII chart below

72 ⇒ H

This process is repeated for all *k* integers which yields the result: HELLO WORLD!

ASCII Chart
This chart contains all the ASCII values and their decimal equivalents that will be used in problem 5. In each pair of columns, the decimal value is on the right and the character is on the left.

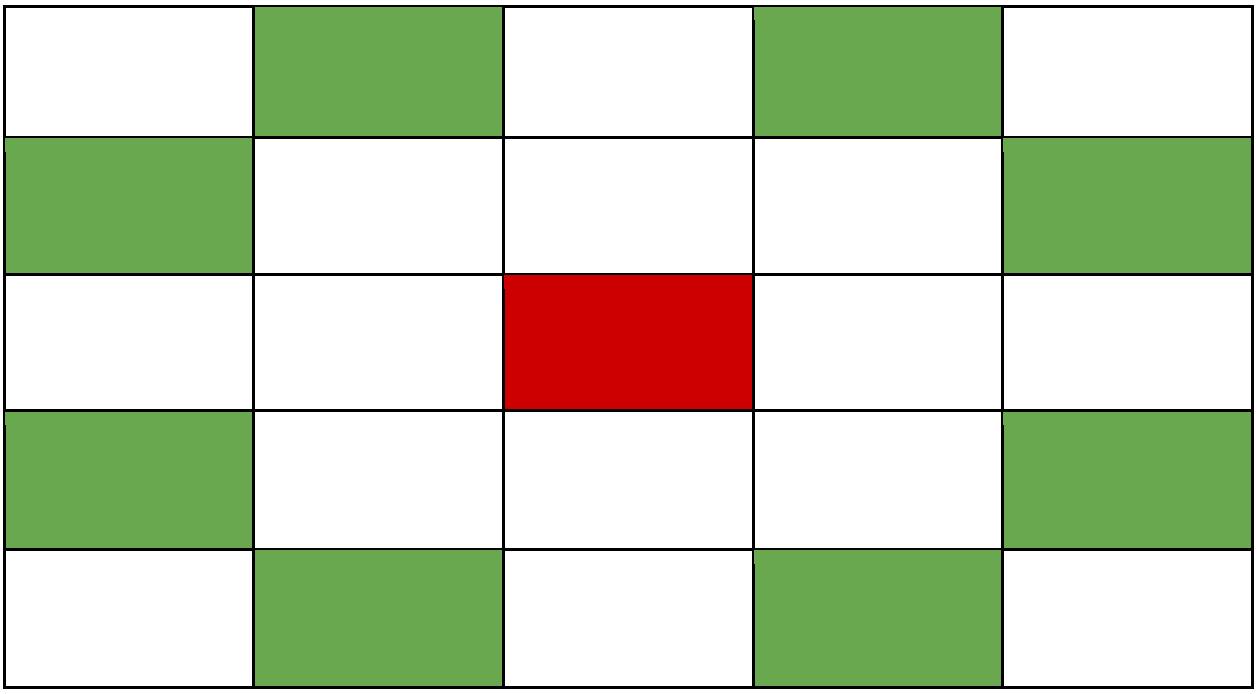
!	33	7	55	H	72	S	83
(	40	8	56	I	73	T	84
)	41	9	57	J	74	U	85
.	46	?	63	K	75	V	86
0	48	A	65	L	76	W	87
1	49	B	66	M	77	X	88
2	50	C	67	N	78	Y	89
3	51	D	68	O	79	Z	90
4	52	E	69	P	80		
5	53	F	70	Q	81		
6	54	G	71	R	82		

Question 7: Robot Takeover!

-

All of Chef Gourmand's waiters have gone missing! He suspects that the evil Maximus the Maitre D' is to blame, but he has no time to confirm his suspicions. In the meanwhile, he decides to use robotic waiters until his normal waiters return.

These robotic waiters follow a certain pathway of movement. They view the restaurant as an *n* by *n* square grid, and are able to move in an "L" shape throughout the grid. The following table shows the possible movements (green) a waiter can make from an initial square (red).



You need to help Chef Gourmand save money by determining, given an initial position and final position, the least amount of moves a waiter needs to make in order to reach the final position from the initial position.

Positions on the grid are marked as follows: Columns and rows are both denoted by numbers where 1 is the first index. Each tile is represented by (column row). For example: "6 7" is the tile found on the 6th column and 7th row.

The first line of input is *n*, the size of the square restaurant grid. The second line of input is the initial position. The third line of the input is the final position. The output should be an integer representing the least number of moves the waiter needs to make, as described above. There could be many possible shortest paths, but only the least number of moves is needed.

We guarantee that 4 <= *n* <=30, and that the initial and final positions are bounded by *n*.

Example 1:
Input:
8
4 5
5 5
Output:
3
Explanation:
The Waiter starts on tile “4 5” and the goal is positioned on tile “5 5”. One possible shortest path is if the waiter moves to square “5 7”, then to square “3 6”. Finally, the waiter moves from “3 6” to “5 5”, making a total of 3 moves.

Recall that we only look for the least number of moves; therefore, although the above path is only one of many, the answer is still correct.

Question 8: Recruit Training

Three chefs are being trained in the kitchen over a period of *n* days.

On each day, there are three kitchen activities to choose from, such as dicing, fileting, and inventory management. Each activity must be done by ONE chef only, and ALL activities must be completed EVERY DAY.

Head Chef Gourmand likes to spice up the scheduling, and so on any given day, one chef is banned from doing a given activity. Help the novice chefs count the number of ways they can plan out their training period. Return the answer modulo (10^9 + 7).

The three chefs are denoted uniquely as 1, 2, and 3. The first line of input is an integer *n*, that represents how many days the three chefs will be in training for. It is guaranteed that *n* < 1000. The next three rows each represent an activity, where the *i*-th element of each row represents the person who is banned from that activity.

Example 1:
Input:
7
1 2 3 3 1 1
2 3 3 1 2 3
2 1 3 1 3 4
Output: 0
Explanation:
We can see that on the third day of activities that the third chef, denoted by 3, is banned from ALL activities. Thus, there is no possible way to complete all activities that day, which would imply that there are ZERO valid ways to plan the training. The output should hence be 0.

Example 2:
Input:
5
1 2 1 3 3
2 1 1 2 1
3 1 3 1 2
Output: 32