Suppandi & Areas

Welcome to Challenge #1 of Daily Challenges March 2020. Today, we are starting off with basic conditions and printing.

Suppandi is trying to take part in the local village math quiz. In the first round, he is asked about shapes and areas. Suppandi, is confused, he was never any good at math. And also, he is bad at remembering the names of shapes. Instead, you will be helping him calculate the area of shapes.

- When he says rectangle he is actually referring to a square.
- When he says square, he is actually referring to a triangle.
- When he says triangle he is referring to a rectangle
- And when he is confused, he just says something random. At this point, all you can do is say 0.

Help Suppandi by printing the correct answer in an integer.

Input Format

The first line contains the number of test cases T. The following T lines contain:

- Name of shape (always in lower case)
- Length of 1 side
- Length of other side

Note: In case of triangle, you can consider the sides as height and length of base

Output Format

- For each test case, print the area of the shape

Sample Input

```
5
triangle
10
20
square
30
40
rectangle
10
10
glass
8
8
cylinder
9
10
```

Output

```
200
600
100
0
0
```

Explanation:

- First line is output of area of rectangle
- Then, output of area of triangle
- Then output of area square
- Finally, something random, so we print 0

Chhota Bheem's Password

It seems many of you helped our Suppandi really well in the previous challenge. He was very impressed, and has referred you to Chhota Bheem. Unlike Suppandi, Chhota Bheem is very smart. He doesn't need your help, instead he wants to test you.

Chhota Bheem's notebook is encrypted with a secret password. He never remembers the password, because he uses a hint and a formula to generate the password.

The hint is a series of N positive numbers. Some of these numbers are purposely left blank, and these are denoted by -1

He tells you that the way to generate the password is:

First you need to fill in the blanks. If numbers on both sides of the blank are both odd, or are both even, then fill it with the absolute difference of both numbers.

However, if both the numbers different (i.e. one is odd and the other is even), in that case fill it with the floor of the arithmetic mean of the 2 numbers.

Second, after you have filled in the blanks, reduce all numbers by 1 except the last number and the numbers whose value is already 1.

Finally, print all the numbers without any spaces. That is the password.

Note: The blanks never appear at the start or end of the list. The blanks also never appear next to each other

Input Format

- The first line contains T the number of test cases
- The following T lines contains N followed by N numbers (all greater than 0). This is the hint as given by Chhota Bheem

Note: N>=3

Output Format

- Print T passwords in a separate line

Example Input

```
3
5 9 -1 8 1 2
3 9 1 4
10 1 22 3 17 -1 5 -1 8 -1 10
```

Example Output

```
87712
814
12121611457110
```

Explanation

1. In the first case, the blank is replaced by (9+8) / 2 = 8.5, since we take floor, we take 8. Then all numbers are subtracted by 1 as per the conditions: 8 7 7 1 2. Finally, print: 87712
2. In the second case, there are no blanks to fill, so we just subtract 1 from applicable numbers. In this case, that is the first number 9. Thus we print 814
3. In the last case, the blanks are filled as follows:

1 22 3 17 12 5 6 8 2 10

Then, subtracting by 1:

1 21 2 16 11 4 5 7 1 10

And printing

12121611457110

Perry the Platypus

Your password cracking abilities were greatly appreciated by Chhota Bheem. Just as he was about to induct you into his team, he got a call from his friend Perry the Platypus. This was quite alarming, because Perry hardly speaks - a phone call is only in emergency situations.

Perry has an important password to crack. Unfortunately, his arch-enemy has hidden this password in a 2D grid of size N x N:

26	28	5	18	27	32	50
13	42	32	11	5	40	27
16	34	23	49	44	17	36
48	10	46	20	13	43	36
1	10	19	29	23	22	20
49	35	15	41	7	29	34
13	23	45	50	31	8	2

These are the instructions given to you by Perry:

1. The highlighted parts above are the generators. The rest of the grid is not useful, only the generators are. The generators are found by starting from bottom left corner, moving to middle of first row and then going to the bottom right corner again. All the while skipping one element. Refer to above image.
2. Thus, you will have N generators
3. You have to collect the N generators, and form a string. Each generator corresponds to an alphabet
4. The alphabet is obtained by from each generator. The number 1 corresponds to a, 2 to b and so on...
5. If we get a number above 26, we will start assigning from a again. So 27 becomes a, 28 is b and so on...
6. Combine the each of the alphabets and the resultant is your answer

Thus, in the above case, we get the following N generators:

13 10 23 18 44 22 2, which translate into: mjwrrvb

Explanation: 13 -> m, 10 -> j, 23 -> w, 18 -> r, 44 -> 18 -> r,
22 -> v, 2 -> b

Given an N and a grid, you must help Perry find the password.

Note: N is always odd and greater than 1. All numbers in the grid are greater than 0.

Input format

- The first line contains T, the number of test cases. Following T lines contain:
- N and N, the size of the 2D grid. This is followed by N x N numbers in a single line

Output format

- Print each password in a single line

Example Input

```
2
3 3 16 13 4 16 13 19 23 8 11
7 7 26 28 5 18 27 32 50 13 42 32 11 5 40 27 16 34 23 49 44 17 36 48 10 46 20 13 43 36 1 10
19 29 23 22 20 49 35 15 41 7 29 34 13 23 45 50 31 8 2
```

Example Output

```
wmk
mjwrrvb
```

Explanation:

The first case forms this grid:

```
16 13 4
16 13 19
23 8 11
```

Generators are 23 13 11 -> wmk

The second case is the same as the one described in the image.

Nobita & Doaremon

Doraemon and Nobita have called upon you. Will you turn them down? No, we wouldn't think so.

It seems Nobita has ingested some poisonous mushrooms (how did he get those mushrooms anyway?) - and has now not able to see clearly. Doraemon is usually very resourceful, but turns our he is allergic to mushrooms.

Nobita has a specific medicine to solve this problem. He tried to say the name of the medicine, but his sentences are jumbled beyond imagination.

Here is how you can help him:

- His sentence is in the form of a very long word. mymedicinesarerighthere
- You need to break this word down in the following way: m, ym ,edi ,cine ,sarer ,ighthe, re
- Then take the first and last characters of the sub-words as long as the sub-word is longer than the previous sub-word: m, ym, ei, ce, sr, ie. Notice we don't touch re because it is shorter than the previous sub-word
- The words on joining gives the name of Nobita's medicines: mymeicesrie

Input Format

- The first line contains T, the number of test cases
- This is followed by T strings (with length >=1)

Output format

- Print the name of Nobita's medicines

Example Input:

```
3
mymedicinesarerighthere
isnobitaalive
testcasesoftenmakemuchmoresensethanthisone
```

Example Output:

```
mymeicesrie
isnoitl
testasofmachent
```

Superman's Encounter

Welcome to a set of new and interesting challenges for the month of April! This month you're going to be on an adventure with superheroes. Fasten your seat-belts, we're in for a bumpy and sometimes time-bending ride.

It's the summer and you are thinking about internship opportunities at some organizations - the Justice League being one of them. You have cleared the first round or tests and are now sitting in a face-to-face interview with Superman.

Superman is planning a journey to his home planet. It is very important for him to know which day he arrives there. They don't follow the 7-day week like us. Instead, they follow a 10-day week with the following days:

Day Number	Name of Day
1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday
8	Kryptonday
9	Coluday
10	Daxamday

Here are the rules of the calendar:

- The calendar starts with Sunday always.
- It has only 296 days. After the 296th day, it goes back to Sunday.

You begin your journey on a Sunday and will reach after n days. You have to tell on which day you will arrive when you reach there.

Input format

- The first line contains T, the number of test cases.
- The following T lines contain a number n (0 < n).

Output format

- For each input, print the name of the day you are arriving on.

Example Input

```
5
1
7
10
300
550
```

Example Output

```
Monday
Kryptonday
Sunday
Thursday
Thursday
```

Explanation

- We arrive 1 day after Sunday, which is day #2 -> Monday
- We arrive 7 days later, which is day #8 -> Kryptonday
- We arrive 10 days later, which means a week later. Thus we arrive at day #11 -> Sunday
- We arrive 300 days later on day # 301. After 295 days of journey we finish the calendar year. We start again from Sunday (297th day) -> Monday (298th day) -> Tuesday (299th day) -> Wednesday (300th day) -> Thursday (301st day)
- We arrive on day # 551. This means we have completed 1 calendar year. In the new year, this is the 255th day. The 250th day is Daxamday, then we have Sunday (251st day) -> Monday (252nd day) -> Tuesday (253rd day) -> Wednesday (254th day) -> Thursday (255th day)

Shaktimaan's Suggestion

You are excited about your internship interview with Superman. It took him a few days to reach home, but it seems he reached on the day that you predicted!

He passes you on the HR interviewer. For the India office, the Head of HR is none other than Shaktimaan!

Shaktimaan is a bit busy putting some prisoners in jail and can't take your interview right away. This is your perfect opportunity to make a good impression. Help him place the prisoners in the respective cells properly.

There are n cells, and Shaktimaan wants to fill p prisoners as follows:

- First all the n cells receive 1 prisoner each.
- Then, each cell from 2 to n receives 1 prisoner each.
- Then, each cell from 3 to n receives 1 prisoner each.
- And so on…
- This keeps on going till we have run out of prisoners. If there are more prisoners left even after filling n rounds, then fill the remaining ones in the 1st cell.

Input Format

- The first line contains T, the number of test cases.
- Each test case contains 2 integers: n (number of cells) and p (number of prisoners).

Note: n <= p

Output Format

- Print the number of prisoners in each cell.

Example Input

```
3
2 3
3 10
3 20
```

Example Output

```
1 2
5 2 3
15 2 3
```

Explanation

- We have 2 cells. We first fill and get 1 1, we have one more prisoner, so we fill 1 2.
- We have 3 cells and 10 prisoners. First we get: 1 1 1, then 1 2 2, then 1 2 3, we have 4 more prisoners left, which we place in the first cell, thus making it 5 2 3.
- Similarly, the last one becomes 15 2 3.

Scarlet Witch's solution

It's not really a daily challenge if you don't crack a password! And you really have a password on your hand this time.

You see, while you got the solution to the prisoner dilemma, Shaktimaan didn't suggest your name for the internship. He wasn't wearing his glasses, and instead drafted you to the forensic team.

It seems one of the prisoners have an ancient lock with them. It is your first day in the forensics team and you want to make a great impression - you decide to unlock this ancient lock.

The lock's password is directly related with the time of the day and a secret series, denoted by the following

$a_n = 2a_{n-1} - 1$

- The hour in current time tells us the value of a_0
- And the minute tells us the number of terms (numbering starts from 0)

Let's say we have the time as 23:05, then:

- a0 = 23
- a1 = 2*23 - 1 = 45
- a2 = 89
- a3 = 177
- a4 = 353
- a5 = 705

Then, we sum up these numbers, we get: 23 + 45 + 89 + 177 + 353 + 705 = 1392

In terms of password, this corresponds to BDJC. (Because 0 corresponds to A, 1 corresponds to B, 2 to C and so on...)

Given the time, please print the password for this lock.

Input format

- The first line contains T, the number of test cases
- Each line after that contains the current time
- Note $a_0 > 0$

Output format

- For each input, print the password

Example Input

```
3
23:05
10:03
01:10
```

Example Output

```
BDJC
BDJ
BB
```

Explanation

1. The sum of the first one becomes: 1392 -> BDJC
2. The sum of second one becomes: 139 -> BDJ
3. The sum of third one becomes: 11 -> BB

Black Widow's Riddle

Good job on cracking that password. Being a superhero isn't easy, so why did you think being a superhero intern would be easy?

Also, did you realize that you applied to the Justice League, yet the last challenge you were working for the Avengers? Black Widow noticed this anomaly and she suspects you aren't from the Avengers - she thinks you're with Thanos!

You want to prove that you're one of the good people. This is the test she has for you. You are given a set of people in a NxN power grid. Some of them are Titans, friends of Thanos.

You can know if they are Titans by knowing if there are any repeated people in a particular row or column of the power grid.

If there are no repeated people in the rows or columns, then the power grid contains no Titans.

Given an NxN grid, you must print:

- SAFE if there are no titans.
- DANGER, followed by number of rows and columns where the repeated people are.

Input Format

- First line contains T, the number of test cases.
- This is followed by N N, the size of the grid followed by N x N numbers denoting the grid.

Output Format

- SAFE if there are no titans.
- DANGER, followed by number of rows and columns where the repeated people are.

Example Input

```
4
2 2 1 2 3 4
3 3 1 2 3 3 1 2 4 5 6
3 3 1 2 3 1 2 3 1 2 3
4 4 1 2 3 1 2 3 4 5 6 7 8 9 10 2 12 1
```

Example Output

```
SAFE
SAFE
DANGER 0 3
DANGER 1 2
```

Explanation

- First and second case are safe because there are no repeated elements.
- In third case, no row has repeated elements but all 3 columns have repeated elements hence DANGER 0 3
- In fourth case, first row as 1 repeated. 2nd column has 2 repeated, and last column also has 1 repeated hence DANGER 1 2

Iron Man's New Invention

Marvel HQ is beginning to warm up to you. They first thought you are an intruder, but now it looks like you're starting to find your way.

You meet Tony Stark, a.k.a Iron Man, on your way out from that dangerous grid. He seems pretty chilled out - he's playing with a new invention a stargazing machine. Seeing as you are the new intern, he wants you to help him with some classification work.

He is looking at some planets, and we want to group planets together by their diameter.

A group's diameter is the maximum sum possible formed by any pair of the planets. For example, if you have the group (2, 4, 3), this group's diameter is 7. This is because the possible diameters are: 2 + 3, 2+ 4, and 4+3, and 4+3 is the maximum from that.

If a certain group contains only 1 planet, then the group's diameter is the same as the diameter of the planet. So a group of (10), has diameter 10.

Given a list of planets, your job is to count minimum number of groups possible, such that their diameter is less than or equal to a certain limit.

If it is not possible, you must print impossible

Input format

1. The first line contains T, the number of test cases
2. In each test case, the first line is N, number of planets and M, the limit of group diameter This is followed by N integers, denoting the diameter for each planet

Note: 0 <= planet_size <= 5000, 1 <= N <= 10000

Example Input

```
3
5 14
6 14 15 0 8
8 5
2 4 3 0 0 0 4 0
10 15
12 9 9 9 7 7 15 5 3 3
```

Output

```
impossible
3
6
```

Explanation

- In the first case, the diameter limit should is 14, but the maximum planet we have is 15. Thus we can't make any groups satisfying the condition
- In the second case, one possible division is:
  Group 1: 4 0 0 0
  Group 2: 3 2 0
  Group 3: 4 0
  There are other combinations too, but we will never get less than 3 groups
- In the 3rd case, one possible division is:
  Group 1: 15
  Group 2: 7 7 5 (here diameter is 14, because maximum sum pair is 14)
  Group 3: 12
  Group 4: 9
  Group 5: 9 3
  Group 6: 9 3

There are other combinations too, but we will never get less than 6 groups.

Coin Change

Write a program to implement coin change making problem i.e. finding the minimum number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

Input Format:

Integer input from stdin.

Output Format:

Print the minimum number of coins required to meet the given target.

Example Input:

16

Output:

4

Explanation:

We need only 4 coins of value 4 each

Example Input:

25

Output:

7

Explanation:

We need 6 coins of 4 value, and 1 coin of 1 value.

Disarium Number

A Number is said to be Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself. Write a program to print number is Disarium or not.

Input Format:

Single Integer Input from stdin.

Output Format:

Yes or No.

Example Input:

175

Output:

Yes

Explanation

11 + 72 +53 = 175

Example Input:

123

Output:

No

Saga of Missing Coin

A Teacher came to the class with a large box that has several coins. Each coin has a number Printed on it. Before Coming to the class, she ensured that all the coins occurs an even number of times. However, while coming to the class one coin fell down and got lost. She wants to find out the number of missing coin.

Input Format:

Take Number from stdin which no of coins n.
Take n-1 array of Integers from stdin.

Output Format:

Print the number of coin which is missed.

Example Input:

```
8
5 7 2 7 5 2 5
```

Output:

```
5
```

Difference Sum

Given a number with maximum of 100 digits as input, find the difference between the sum of odd and even position digits.

Input Format:

Take a number in the form of String from stdin.

Output Format:

Print the difference between sum of even and odd digits

Example input:

1453

Output:

1

Explanation:

Here, sum of even digits is 4 + 3 = 7
sum of odd digits is 1 + 5 = 6.

Difference is 1.

Note that we are always taking absolute difference.

Check Product of Digits

Write a code to check whether product of digits at even places is divisible by sum of digits at odd place of a positive integer.

Input Format:

Take an input integer from stdin.

Output Format:

Print TRUE or FALSE.

Example Input:

1256

Output:

TRUE

Example Input:

1595

Output:

FALSE

Bucket IDs

A data company wishes to store its data files on the server. They wish to store N files. Each files has a particular size. The server stores the files in buckets. The bucket ID is calculated as the sum of the digits of its file size. The server returns the bucket ID for every file request where the file is stored.

Write an algorithm to find the bucket IDs where the files are stored.

Constraints

- 0<=numFiles<=1e6
- 0<=fileSize[i]<=1e6
- 0<=i<numFiles

Input

The first line of the input consists of an integer numFiles, representing the number of files to be stored (N).

The second line consists of N space-separated integers - fileSize[i] , where fileSize[i] represents the size of the files.

Output

Print N space-separated integers representing the bucket IDs for each file, respectively.

Example Input

```
3
41 467 334
```

Output

```
5 17 10
```

Check whether a number is an Automorphic number or not

An automorphic number is a number whose square ends with the number itself.

For example, 5 is an automorphic number because 5*5 =25. The last digit is 5 which same as the given number.

If the number is not valid, it should display “Invalid input”.

If it is an automorphic number display “Automorphic” else display “Not Automorphic”.

Input Format:

Take a Integer from Stdin

Output Format:

Print Automorphic if given number is Automorphic number, otherwise Not Automorphic

Example input:

5

Output:

Automorphic

Example input:

25

Output:

Automorphic

Example input:

7

Output:

Not Automorphic

Abundant Number

An abundant number is a number for which the sum of its proper divisors is greater than the number itself.

Proper divisors of the number are those that are strictly lesser than the number.

Input Format:

Take input an integer from stdin

Output Format:

Print Yes if given number is Abundant. Otherwise, print No

Example input:

12

Output:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is 1 + 2 + 3 + 4 + 6 = 16. Since sum of proper divisors is greater than the given number, 12 is an abundant number.

Example input:

13

Output:

No

Explanation

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater than the given number, 13 is not an abundant number.

Christmas Discount

An e-commerce company plans to give their customers a special discount for Christmas. They are planning to offer a flat discount. The discount value is calculated as the sum of all the prime digits in the total bill amount.

Write an algorithm to find the discount value for the given total bill amount.

Constraints

1 <= orderValue < 10e100000

Input

The input consists of an integer orderValue, representing the total bill amount.

Output

Print an integer representing the discount value for the given total bill amount.

Example Input

578

Output

12

Coin denominations

Write a program to take value V and we want to make change for V Rs, and we have infinite supply of each of the denominations in Indian currency, i.e., we have infinite supply of {1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and/or notes needed to make the change.

Input Format:

Take an integer from stdin.

Output Format:

print the integer which is change of the number.

Example Input:

64

Output:

4

Explanation:

We need a 50 Rs note and a 10 Rs note and two 2 rupee coins.

Example Input:

49

Output:

5

Explanation:

We need a two 20 Rs notes and a 5 Rs coins and two 2 rupee coins.

Generate Password

Given input of array of string in format <employee name> <employee number> separated by commas, you have to generate password for each employee.

Employee name contains only alphabets and employee number contains only digits.

The conditions to generate the password are

- The password will be single the character in the name of the employee at the index k
- where k is the digit that is present in the employee number that is less than or equal to the length of the employee name.

Note: The string index should be considered from 1.

Example

Input:

Robert:36787,Tina:68721,Jo:56389

Output:

tiX

Explanation

Length of Robert is 6 and 6 is present in employee number of Robert 36787, so return the alphabet at position 6 that is 't'.
Length of Tina is 4 and 4 is not present in the 68721 so select the number which is max and less than the length of Tina so select 2 return the alphabet at position 2 that is 'i'.
Length of Jo is 2 it is not present in 56389 and there is not present any number which is less than 2 so return 'X'.

Constraint

1<length(employee Name)<10

Input Format

A single string that that has sub strings separated by commas.

- employee name and employee number were separated by colon

Output Format

A single string formed by concatenating the of passwords of all employees.

Ugly Numbers

A number is considered to be ugly if its only prime factors are 2, 3 or 5.

[1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, …] is the sequence of ugly numbers.

Task:

Create a program which takes a number n as input and checks if it's an ugly number.

Print ugly if it is ugly, else print not ugly

Hint:

An ugly number U can be expressed as: U = 2^a * 3^b * 5^c, where a, b and c are non-negative integers.

Input Format

First line contains number tc showing the number of test cases.
The following tc lines contains the number n which you need to check whether it is ugly or not.

Output Format

The output contains tc number of lines.
Where each line contains the "ugly" or "not ugly" as per the above mentioned rules.

Simple Chessboard

Write a program that prints a simple chessboard.

Input format:

The first line contains the number of inputs T.

The lines after that contain a different value for size of the chessboard

Output format:

Print a chessboard of dimensions’ size * size. Print W for white spaces and B for black spaces.

Input:

```
2
3
5
```

Output:

```
WBW
BWB
WBW
WBWBW
BWBWB
WBWBW
BWBWB
WBWBW
```

Print Our Own Chessboard

Our Own Chessboard

Let’s print a chessboard!

Write a program that takes input:

The first line contains T, the number of test cases
Each test case contains an integer N and also the starting character of the chessboard

Output Format

Print the chessboard as per the given examples

Sample Input / Output

Input:

```
2
2 W
3 B
```

Output:

```
WB
BW
BWB
WBW
BWB
```

Zombie World

One-day Bob is playing Zombie World video game. In Zombie World game each round will contain N zombie’s and each zombie’s energy is Zi (where 1 <= i <= N).

Bob will start the round with B energy. In order to move to the next level Bob need to kill all the N zombie’s but Bob can select any one among N Zombies. If energy of Bob (B) is less than Zombie energy (Zi) then Bob will die and lose the round else Bob will won, during the fighting with zombie, Bob will lose his energy by (Zi % 2) + (Zi / 2). At any point of game Bob will play optimally. Now your task is to find out whether Bob can reach to the next level or not.

Input Format

First line will contains B and N, separated by space, where B is the energy of Bob and N is the number of Zombie. Next line will contain N spaced integers each will represent the energy of zombie.

Line 1: B N, where B is the energy of Bob and N is the number of Zombie
Line 2: Zi, where Zi is a list containing energy of zombies separated by space

Constraints

1 <= N <= 10^4
1 <= B <= 10^9
1 <= Zi <= 10^5

Note: for this problem all the divisions are integer divisions.

Output Format:

Print YES or NO depending upon whether Bob can reach the next level or not.

For Valid Input, print

YES Or NO

For Invalid Input, print

Invalid Input

Sample Input / Output

Input

35 3

5 9 6

Output

YES

Input

456 68

a

Output

Invalid Input

Input

4 4

1 3 2 4

Output

NO

Stone Game-One Four

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins. They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

Input Format

First line starts with T, which is the number of test cases. Each test case will contain N number of stones.

Output Format

Print "Yes" in the case Alice wins, else print "No".

Constraints

1<=T<=1000
1<=N<=10000

Sample Input and Output

Input

```
3
1
6
7
```

Output

```
Yes
Yes
No
```

Distribute Books

For enhancing the book reading, school distributed story books to students as part of the Children's day celebrations.

To increase the reading habit, the class teacher decided to exchange the books every weeks so that everyone will have a different book to read. She wants to know how many possible exchanges are possible.

If they have 4 books and students, the possible exchanges are 9. Bi is the book of i-th student and after the exchange he should get a different book, other than Bi.

B1 B2 B3 B4 - first state, before exchange of the books
B2 B1 B4 B3
B2 B3 B4 B1
B2 B4 B1 B3
B3 B1 B4 B2
B3 B4 B1 B2
B3 B4 B2 B1
B4 B1 B2 B3
B4 B3 B1 B2
B4 B3 B2 B1

Find the number of possible exchanges, if the books are exchanged so that every student will receive a different book.

Constraints

1<= N <= 1000000

Input Format

Input contains one line with N, indicates the number of books and number of students.

Output

Output the answer modulo 1000000007.

Test Case Explanation

Example 1 Input

4

Output

9

Intersection of 2 Sorted Arrays

Find the intersection of two sorted arrays.
OR in other words,
Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

```
1
3 10 17 57
6 2 7 10 15 57 246
```

Output:

```
10 57
```

Input:

```
1
7 1 2 3 3 4 5 6
2 1 6
```

Output:

```
1 6
```

Check pair with difference k

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn’t.

Example

Input

```
1
3 1 3 5
4
```

Output:

```
1
```

Input

```
1
3 1 3 5
99
```

Output

```
0
```

Input

```
4
3 1 3 5
4
3 1 3 5
99
6 2 7 10 15 57 246
47
6 2 7 10 15 57 246
999
```

Output

```
1
0
1
0
```

Collecting Candies

Krishna loves candies a lot, so whenever he gets them, he stores them so that he can eat them later whenever he wants to.

He has recently received N boxes of candies each containing Ci candies where Ci represents the total number of candies in the ith box. Krishna wants to store them in a single box. The only constraint is that he can choose any two boxes and store their joint contents in an empty box only. Assume that there are infinite number of empty boxes available.

At a time, he can pick up any two boxes for transferring and if both the boxes say contain X and Y number of candies respectively, then it takes him exactly X+Y seconds of time. As he is to eager to collect all of them he has approached you to tell him the minimum time in which all the candies can be collected.

Input Format

- First line of input is number of test case T
- Each test case is comprised of two inputs
    - First input of a test case is the number of boxes N
    - Second input is N integers delimited by whitespace denoting number of candies in each box

Output Format:

Print minimum time required, in seconds, for each of the test case. Print each output on a new line.

Constraints:

1. 1 ≤T≤10
2. 1 ≤N≤ 10000
3. 1 ≤ [Candies in each box] ≤ 100009

Sample Input and Output

Input

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

Output

```
19
```

Explanation

4 boxes, each containing 1, 2, 3 and 4 candies respectively.

Adding 1 + 2 in a new box takes 3 seconds

Adding 3 + 3 in a new box takes 6 seconds

Adding 4 + 6 in a new box takes 10 seconds

Hence total time taken is 19 seconds. There could be other combinations also, but overall time does not go below 19 seconds.

Input

```
1
5
1 2 3 4 5
```

Output

```
33
```

Explanation

5 boxes, each containing 1, 2, 3, 4 and 5 candies respectively.

Adding 1 + 2 in a new box takes 3 seconds

Adding 3 + 3 in a new box takes 6 seconds

Adding 4 + 5 in a new box takes 9 seconds

Adding 6 + 9 in a new box takes 15 seconds

Hence total time taken is 33 seconds. There could be other combinations also, but overall time does not go below 33 seconds.

Bottle Necks

There are N bottles. i^{th} bottle has A[i] radius. Once a bottle is enclosed inside another bottle, it ceases to be visible. Minimize the number of visible bottles.

You can put i^{th} bottle into j^{th} bottle if following condition is fulfilled:

- i^{th} bottle itself is not enclosed in another bottle.
- j^{th} bottle does not enclose any other bottle.
- Radius of bottle i is smaller than bottle j (i.e. A[i] < A[j]).

Constraints

- 1 <= N <= 100000
- 1 <= A[i] <= 10^{18}

Input Format

- First line contains T, the number of test cases. For each test case:
- First line contains a single integer N denoting the number of bottles.
- Second line contains N space separated integers, i^{th} integer denoting the radius of i^{th} bottle.
- (1 <= i <= N)

Output

- Minimum number of visible bottles after all the operations.

Example Input

```
2
8
1 1 2 3 4 5 5 4
5
5 7 1 2 4
```

Output

```
2
1
```

Explanation

In the first case:

- 1st bottle can be kept in 3rd one 1-->2, which makes following bottles visible [1,2,3,4,5,5,4].
- Similarly, after following operations, the following will be the corresponding visible bottles.

Operation	Visible Bottles
2 --> 3	[1,3,4,5,5,4]
3 --> 4	[1,4,5,5,4]
4 --> 5	[1,5,5,4]
1 --> 4	[5,5,4]
4 --> 5	[5,5]

- Finally, there are 2 bottles which are visible. Hence, the answer is 2.

In the second case, we can follow the operations: 1 -> 2 -> 4 -> 5 -> 7.

Chakravyuha

During the battle of Mahabharat, when Arjuna was far away in the battlefield, Guru Drona made a Chakravyuha formation of the Kaurava army to capture Yudhisthir Maharaj. Abhimanyu, young son of Arjuna was the only one amongst the remaining Pandava army who knew how to crack the Chakravyuha. He took it upon himself to take the battle to the enemies.

Abhimanyu knew how to get power points when cracking the Chakravyuha. So great was his prowess that rest of the Pandava army could not keep pace with his advances. Worried at the rest of the army falling behind, Yudhisthir Maharaj needs your help to track of Abhimanyu's advances. Write a program that tracks how many power points Abhimanyu has collected and also uncover his trail

A Chakravyuha is a wheel-like formation. Pictorially it is depicted as below:

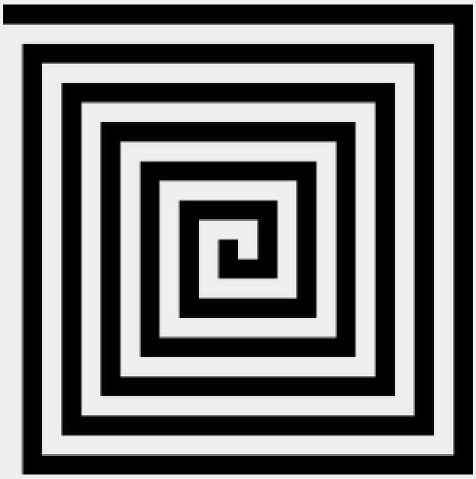
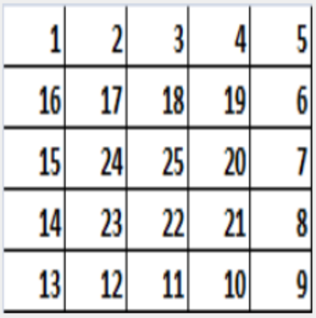


Fig 1. Chakravyuha

A Chakravyuha has a very well-defined co-ordinate system. Each point on the co-ordinate system is manned by a certain unit of the army. The Commander-In-Chief is always located at the center of the army to better co-ordinate his forces. The only way to crack the Chakravyuha is to defeat the units in sequential order.

A Sequential order of units differs structurally based on the radius of the Chakra. The radius can be thought of as length or breadth of the matrix depicted above. The structure i.e. placement of units in sequential order is as shown below:



1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

Fig 2. Army unit placements in Chakravyuha of size 5

The entry point of the Chakravyuha is always at the (0,0) co-ordinate of the matrix above. This is where the 1st army unit guards. From (0,0) i.e. 1st unit Abhimanyu has to march towards the center at (2,2) where the 25th i.e. the last of the enemy army unit guards. Remember that he has to proceed by destroying the units in sequential fashion. After destroying the first unit, Abhimanyu gets a power point. Thereafter, he gets one after destroying army units which are multiples of 11. You should also be a in a position to tell Yudhisthir Maharaj the location at which Abhimanyu collected his power points.

Input Format

First line of input will be length as well as breadth of the army units, say N

Output Format

- Print NxN matrix depicting the placement of army units, with unit numbers delimited by (\t) Tab character
- Print Total power points collected
- Print coordinates of power points collected in sequential fashion (one per line)

Constraints

0 < N <=100

Sample Input & Output

Input

```
2
```

Output

```
1 2
4 3
Total Power points : 1
(0,0)
```

Input

```
5
```

Output

```
1 2 3 4 5
16 17 18 19 6
15 24 25 20 7
14 23 22 21 8
13 12 11 10 9
Total Power points : 3
(0,0)
(4,2)
(3,2)
```

Matrix Rotations

You are given a square matrix of dimension N. Let this matrix be called A. Your task is to rotate A in clockwise direction by S degrees, where S is angle of rotation. On the matrix, there will be 3 types of operations viz.

1. Rotation
   Rotate the matrix A by angle S, presented as input in form of A S

2. Querying
   Query the element at row K and column L, presented as input in form of Q K L

3. Updation
   Update the element at row X and column Y with value Z, presented as input in form of U X Y Z

Print the output of individual operations as depicted in Output Specification.

Input Format:

Input will consist of three parts, viz.

1. Size of the matrix (N)
2. The matrix itself (A = N * N)
3. Various operations on the matrix, one operation on each line. (Beginning either with A, Q or U)
4. -1 will represent end of input

Note:

- Angle of rotation will always be multiples of 90 degrees only.
- All Update operations happen only on the initial matrix. After update all the previous rotations have to be applied on the updated matrix

Output Format:

For each Query operation print the element present at K-L location of the matrix in its current state.

Constraints:

1<=N<=1000
1<=Aij<=1000
0<=S<=160000
1<=K, L<=N
1<=Q<=100000

Sample Input and Output

Input

```
2
1 2
3 4
A 90
Q 1 1
Q 1 2
A 90
Q 1 1
U 1 1 6
Q 2 2
-1
```

Output

```
3
1
4
6
```

Explanation:

Initial Matrix

```
1 2
3 4
```

After 90-degree rotation, the matrix will become

```
3 1
4 2
```

Now the element at A_{11} is 3 and A_{12} is 1.

Again the angle of rotation is 90 degrees, now after the rotation the matrix will become

```
4 3
2 1
```

Now the element at A_{11} is 4.

As the next operation is Update, update initial matrix i.e.

```
6 2
3 4
```

After updating, apply all the previous rotations (i.e. 180 = two 90 degree rotations)

The matrix will now become

4 3
2 6

Now A_{22} is 6.

Super ASCII String Checker

In the Byteland country a string S is said to super ASCII string if and only if count of each character in the string is equal to its ASCII value.

In the Byteland country ASCII code of a is 1, b is 2 ...z is 26.

Your task is to find out whether the given string is a super ASCII string or not.

Input Format:

First line contains number of test cases T, followed by T lines, each containing a string S.

Output Format:

For each test case print Yes if the String S is super ASCII, else print No

Constraints

1<=T<=100
1<=|S|<=400, S will contains only lower case alphabets ('a'-'z')

Sample Input and Output

Input

```
2
bba
scca
```

Output

```
Yes
No
```

Explanation

In case 1, viz. String "bba" -
The count of character 'b' is 2. Ascii value of 'b' is also 2.
The count of character 'a' is 1. Ascii value of 'a' is also 1.
Hence string "bba" is super ascii.

Cyclic Palindrome

A string is said to be palindrome, if it reads the same from both the ends. Given a string S, you are allowed to perform cyclic shifts. More formally, you can pick any one character from any end (head or tail) and you can append that character at the other end. For example, if the string is abc, then if we do a shift using the character at head position then the string becomes bca. Similarly, if we do the shift using the character at the tail then the input string becomes cab. Your task is to find out the minimum number of shifts needed to make the given string, a palindrome.

In case, we can't convert the string to palindrome then print -1.

Input Format

First line starts with T i.e. number of test cases, and then T lines will follow each containing a string S.

Output Format

Print the minimum number of cyclic shifts for each string if it can be made a palindrome, else -1.

Constraints

1<=T<=100
1<=|S|<=300, S will contains only lower case alphabets a-z.

Sample Input and Output

Input

```
4
abbb
aaabb
aabb
abc
```

Output

```
-1
1
1
-1
```

Explanation:

For Test Case 2 (aaabb):

Shift the character at the tail to the head and the result will be baaab, which is a palindrome. This is an operation which requires minimum number of shifts to make the given string a palindrome.

For Test Case 3 (aabb):

One way to convert the given string to palindrome is, shift the character at the head to the tail, and the result will be abba, which is a palindrome. Another way is to shift the character at the tail to the head, and the result will be baab, which is also a palindrome. Both require only one shift.

String Rotation

Rotate a given String in the specified direction by specified magnitude.

After each rotation make a note of the first character of the rotated String, After all rotation are performed the accumulated first character as noted previously will form another string, say FIRSTCHARSTRING.

Check If FIRSTCHARSTRING is an Anagram of any substring of the Original string.

If yes print YES otherwise NO.

Input

- The first line contains the original string s.
- The second line contains a single integer q.
- The ith line of the next q lines contains character d[i] denoting direction and integer r[i] denoting the magnitude.

Output

YES or NO

Constraints

1 <= Length of original string <= 30
1<= q <= 10

Sample Input & Output

Input

carrace
3
L 2
R 2
L 3

Output

NO

Explanation

After applying all the rotations, the FIRSTCHARSTRING string will be rcr which is not anagram of any sub string of original string carrace

Ramanujan Numbers and The Taxicab Problem

If you mention the number “1729” or the phrase “Taxicab Problem” to any mathematician, it will immediately bring up the subject of the self-taught Indian mathematical genius Srinivasa Ramanujan. When Ramanujan was dying of tuberculosis in a hospital, G. H. Hardy would frequently visit him. It was on one of these visits that the following occurred according to C. P. Snow.

> “Hardy used to visit him, as he lay dying in hospital at Putney. It was on one of those visits that there happened the incident of the taxicab number. Hardy had gone out to Putney by taxi, as usual his chosen method of conveyance. He went into the room where Ramanujan was lying. Hardy, always inept about introducing a conversation, said, probably without a greeting, and certainly as his first remark: ‘I thought the number of my taxicab was 1729. It seemed to me rather a dull number.’ To which Ramanujan replied: ‘No, Hardy! No, Hardy! It is a very interesting number. It is the smallest number expressible as the sum of two cubes in two different ways.’”

Since then, integer solutions to:

I^3 + J^3 = K^3 + L^3

have been called “Ramanujan Numbers”.

The first 2 of these are:

1729 (1^3 + 12^3, 9^3 + 10^3)
4104 (2^3 + 16^3, 9^3 + 15^3)

Given a number, write a program to find whether it is a Ramanujam Number or not.

Input and Output Format:

Input consists of a single integer.
Output yes if it is a Ramanujam Number. Output no otherwise.

Sample Input 1:

1729

Sample Output 1:

yes

Sample Input 2:

1854

Sample Output 2:

no