Tilling Problem - II

Given a floor of size **n x m**. Find the number of ways to tile the floor with tiles of size **1 x m** . A tile can either be placed horizontally or vertically.

Input Format:

First line of input contains an integer **T** denoting the number of test cases. Then T test cases follow.   
The first line of each test case contains two integers **N** and **M**.

Constraints:

1 <= T<= 1000 1 <= N,M <= 100000

output format

Print answer for every test case in a new line modulo **10^9+7**.

sample input

2

2 3

4 4

sample output

1

2

Generate Parentheses

Given an integer 'n'. Print all the possible pairs of 'n' balanced parentheses.  
The output strings should be printed in the sorted order considering '(' has higher value than ')'.

Input Format:

Single line containing an integral value 'n'.

Constraints:

1<=n<=11

output format

Print the balanced parentheses strings with every possible solution on new line.

sample input

2

sample output

()()

(())

Count Number of Binary Strings

You are provided an integers N. You need to count all possible distinct binary strings of length N such that there are no consecutive 1’s.

Input Format:

First line contains integer t which is number of test case. For each test case, it contains an integer n which is the the length of Binary String.

Constraints:

1<=t<=100 1<=n<=90

output format

Print the number of all possible binary strings.

sample input

2

2

3

sample output

3

5

Friends Pairing Problem

Given n friends, each one can remain single or can be paired up with some other friend. Each friend can be paired only once. Find out the total number of ways in which friends can remain single or can be paired up.

Input Format:

First line contains an integer t denoting the number of test cases. Next t lines contain an integer n each.

Constraints:

1<= n < 30

output format

Output t lines each line describing the answer.

sample input

1

3

sample output

4

Recursion-Codes of the string

Take as input str, a string. Assume that value of a=1, b=2, c=3, d=4, …. z=26. Write a recursive function (return type Arraylist) to print all possible codes for the string. E.g. for “1123” possible codes are aabc, kbc, alc, aaw, kw.

Input Format:

Enter a number

Constraints:

None

output format

Display all the possible codes

sample input

1125

sample output

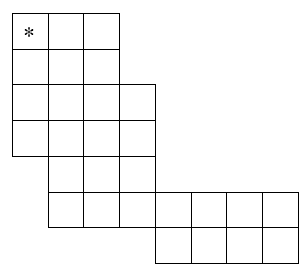
[aabe, aay, ale, kbe, ky]

C++

Funky Chess Board

A knight is a piece used in the game of chess. The chessboard itself is square array of cells. Each time a knight moves, its resulting position is two rows and one column, or two columns and one row away from its starting position. Thus a knight starting on row r, column c – which we’ll denote as (r,c) – can move to any of the squares (r-2,c-1), (r-2,c+1), (r-1,c-2), (r-1,c+2), (r+1,c-2), (r+1,c+2), (r+2,c-1), or (r+2,c+1). Of course, the knight may not move to any square that is not on the board.

Suppose the chessboard is not square, but instead has rows with variable numbers of columns, and with each row offset zero or more columns to the right of the row above it. The figure to the left illustrates one possible configuration. How many of the squares in such a modified chessboard can a knight, starting in the upper left square (marked with an asterisk), not reach in any number of moves without resting in any square more than once? Minimize this number.



If necessary, the knight is permitted to pass over regions that are outside the borders of the modified chessboard, but as usual, it can only move to squares that are within the borders of the board.

Input Format:

First line contains an integer ***n***, representing the side of square of chess board. The next ***n*** line contains ***n*** integers separated by single spaces in which jjth integer is 1 if that cell(i,j) is part of chessboard and 0 otherwise.

Constraints:

The maximum dimensions of the board will be 10 rows and 10 columns. That is, any modified chessboard specified by the input will fit completely on a 10 row, 10 column board.

output format

Print the minimum number of squares that the knight can not reach.

sample input

3

1 1 1

1 1 1

1 1 1

sample output

1

Class Assignment

In a mathematics class, Teacher ask Alice to find the number of all n digit distinct integers which is formed by the two distinct digits **ai** and **bi** but there is a rule to form n digit integer.

**Rule:** She has to form n digit integer by using two digits ai and bi without consecutive bi.

Alice is very weak in maths section. Help her to find the number of such n digit integers.

Input Format:

The first line contains T, the number of test cases. Further T lines contains the value n which is the number of digit in the integer.

Constraints:

1<=T<=40 1<=n<44

output format

For each test case print the number of such n digit integer.

sample input

3

1

2

3

sample output

#1 : 2

#2 : 3

#3 : 5

Smart Keypad - I

You will be given a numeric string **S**. Print all the possible codes for **S**.

Following vector contains the codes corresponding to the digits mapped.

string table[] = { " ", ".+@$", "abc", "def", "ghi", "jkl" , "mno", "pqrs" , "tuv", "wxyz" };

For example, string corresponding to **0** is " " and **1** is ".+@$"

Input Format:

A single string containing numbers only.

Constraints:

length of string <= 10

output format

All possible codes one per line in the following order.

The letter that appears first in the code should come first

sample input

12

sample output

.a

.b

.c

+a

+b

+c

@a

@b

@c

$a

$b

$c

Smart Keypad - Advanced

Given a long vector of strings, print the strings that contain the strings generated by numeric string **str**.

string searchIn [] = {

"prateek", "sneha", "deepak", "arnav", "shikha", "palak",

"utkarsh", "divyam", "vidhi", "sparsh", "akku"

};

For example, if the input is **26** and the string is **coding**, then output should be **coding** since **26** can produce **co** which is contained in **coding**.

Input Format:

A numeric string **str**

Constraints:

len(str) < 10 No of strings in the vector < 10

output format

Each matched string from the given vector.

sample input

34

sample output

vidhi

divyam

sneha

Tricky Permutations

Given a string containing duplicates, print all its distinct permutations such that there are no duplicate permutations and all permutations are printed in a lexicographic order.

**NOTE: DO NOT USE MAP OR SET.**

Input Format:

The first and only line of the test case contains the input string.

Constraints:

Length of the string <= 8

output format

Print all the distinct permutations in a lexicographic order such that each permutation is in a new line. Note that there should not be any duplicate permutations.

sample input

ABA

sample output

AAB

ABA

BAA

No same Permutations

Given a collection of numbers containing duplicates, print all possible permutations for the given collection of numbers in a lexicographic manner. Make sure that no two permutations are same.   
**NOTE: Try doing it without using SET or MAP!**

Input Format:

The first line will contain the number of elements, **N** in the array.  
The second line will contain N elements of the array, **A[i]**.

Constraints:

1<= N <= 10 0 <= A[i] <= 100000

output format

Print all unique lexicographic permutations of the array. Each permutation will contain space separated elements. The next permutation should be printed in a new line.

sample input

3

1 1 3

sample output

1 1 3

1 3 1

3 1 1

Sum It Up

You are given an array of numbers and a target number(**T**) , print all unique combinations in the array whose sum equals the target number **T**. Note that each number in the array may only be used once in the combination.  
**Note:**

* All numbers (including target) will be positive integers
* Elements in the combination must be in non-descending order
* There should be no duplicate combinations

Input Format:

The first line will contain **N** indicating the number of elements in the array.  
The second line will contain space separated **N** integers , **A[i]**.   
The third line will contain the target number **T**.

Constraints:

N <= 15 1<= A[I] <= 100

output format

Print all unique combinations of the numbers satisfying the above constraints in a separate line in lexicographic manner.

sample input

7

10 1 2 7 6 1 5

8

sample output

1 1 6

1 2 5

1 7

2 6

Expand the Pond

You are given a matrix with **N** rows and **M** columns. Each cell is either dry or has water. We say that two cells are neighbours if they share one of their four sides. A pond is a maximal subset of cells containing water such that any cell is accessible from any other cell by moving only along neighbours. You should add water to exactly one cell in order to maximise the size of the pond.

Input Format:

The first line contains two integers **N** and **M**. Each of the next **N** lines contains **M** integers: **0** for a dry cell and **1** for a cell containing water.

Constraints:

1 ≤ N,M ≤ 1000 There is at least one dry cell and one cell filled with water in the matrix.

output format

Print a single integer representing the largest possible size of the pond.

sample input

3 3

0 1 1

0 0 1

0 1 0

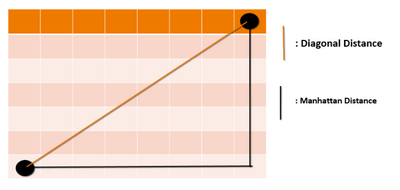
sample output

5

Minimum Time Traversal

**Distance:** The distance between two points in a grid based on a strictly horizontal and/or vertical path (i.e along the grid lines), as opposed to the **Manhattan Distance** or **Diagonal Distance**.

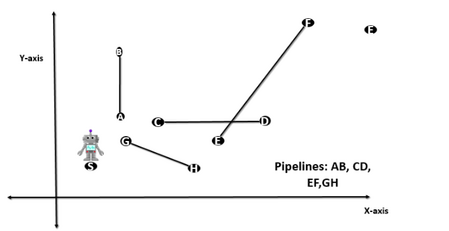
The **Manhattan Distance** is the simple sum of the horizontal and vertical components, where as the **Diagonal Distance** might be computed by applying the **Pythagorean Theorem**.



**Problem Statements:** A robot is moving on co-ordinate axis. Time taken to move from one point to another point is equal to the Manhattan Distance.

Ex. Time taken to move from point ( x1 , y1) to ( x2 , y2) is |x1 - x2| + |y1 - y2| where | a | is equal to modulus function of a.

Starting Point (S) and Ending Point (E) of the robot is fixed. There are n wormhole like pipelines also on the coordinate axis. Time taken to move from one point to another point of each pipeline is given.



Robot can use those pipelines to move from statring point to ending point. Can you help to find the minimum time required during traversal.

Input Format:

The first line contains T, the number of test cases.

The description of T test cases follows.

The first line of each test case contains integer n which represents the number of pipelines.

The next line contains four space seperated integer. The first two integer is the x and y coordinate of starting point and last two integer is the x and y coordinate of the ending points. The next n lines contains 5 space-separated character x1 , y1, x2 , y2, t . The first four integers are the x and y co-ordinate of terminal of the pipelines and 5 th integer is time to cross that pipelines .

Constraints:

1 <= T <= 30 0 <= n <= 5 1 <= x,y <= 2000

output format

For each test case you have to print the output in this format (#Test Case Number : minimum time taken by robot to traverse from begining to ending point.)

sample input

3

0

20 20 100 100

1

20 20 100 100

25 25 30 30 5

3

20 20 100 100

35 35 50 50 0

30 30 25 25 0

40 40 60 60 100

sample output

#1 : 160

#2 : 155

#3 : 120

Subset Sum Easy

Mike is a very passionate about sets. Lately, he is busy solving one of the problems on sets. He has to find whether if the sum of any of the non-empty subsets of the set A is zero.

Input Format:

The first line contains an integer T, which is the total number of test cases. T

test cases follow.

Each test case consists of two lines.

The first line consists of a single integer N, which is the number of elements

present in the set A.

The second line contains the integer in the set.

Constraints:

1 ≤ T ≤10 1 ≤ N ≤ 4 -10^5 ≤ A[i] ≤ 10^5

output format

If the sum of any of the subset is zero, then print “Yes” (without

quotes) else print “No”(without quotes).

sample input

1

4

1 2 3 -3

sample output

Yes

Optimal Game Strategy-I

Piyush and Nimit are playing a coin game. They are given n coins with values x1, x2 …. xn where 'n' is always even. They take alternate terms. In each turn, a player picks either the first coin or the last coin from the row and removes it from the row. The value of coin is received by that player. Determine the maximum value that piyush can win if he moves first. Both the players play optimally.

Input Format:

First line contains the number of coins 'n'. Second line contains n space separated integers which is the value of ith coin.

Constraints:

N < 30

output format

Print a single line with the maximum possible value.

sample input

4

1 2 3 4

sample output

6

Optimal Game Strategy-II

Piyush and Nimit are playing a coin game. They are given n coins with values x1, x2 …. xn where 'n' is always even. They take alternate terms. In each turn, a player picks either the first coin or the last coin from the row and removes it from the row. The value of coin is received by that player. Determine the maximum value that piyush can win if he moves first. Both the players play optimally.

Input Format:

First line contains the number of coins 'n'. Second line contains n space separated integers which is the value of ith coin.

Constraints:

N <= 10000

output format

Print a single line with the maximum possible value.

sample input

4

1 2 3 4

sample output

6

Recursion-Dictionary Order(Larger)

Take as input str, a string. Write a recursive function which prints all the words possible by rearranging the characters of this string which are in dictionary order larger than the given string

Input Format:

Enter a string

Constraints:

None

output format

Display all the words larger than the string entered in separate lines

sample input

cab

sample output

cba

N\_Knight Problem

Take as input N, the size of a chess board. We are asked to place N number of Knights in it, so that no knight can kill other.

a. Write a recursive function which returns the count of different distinct ways the knights can be placed across the board. Print the value returned.

b.Write a recursive function which prints all valid configurations (void is the return type for function).

Input Format:

Enter the size of the chessboard N

Constraints:

None

output format

Display the number of ways a knight can be placed and print all the possible arrangements in a space separated manner

sample input

2

sample output

{0-0} {0-1} {0-0} {1-0} {0-0} {1-1} {0-1} {1-0} {0-1} {1-1} {1-0} {1-1}

6

N-Queen Hard

You are given an empty chess board of size N\*N. Find the number of ways to place N queens on the board, such that no two queens can kill each other in one move. A queen can move vertically, horizontally and diagonally.

Input Format:

A single integer N, denoting the size of chess board.

Constraints:

1 ≤ N < 15

output format

A single integer denoting the count of solutions.

sample input

4

sample output

2

CROSSWORD

You are given a dictionary consisting of **N** words each containing **lowercase** letters 'a'-'z'. You also have **MxM** crossword consisting of lowercase letters 'a'-'z'. You want to print all possible words that can be formed by a sequence of adjacent characters. Note that you can move to any of **8 adjacent** characters, but a word should **not** have multiple instances of same cell. **Note :- if a word is present in crossword more than once print it once.**

Input Format:

First line contains integer N. Then follow N space seperated words. Next line contains integer M. Then you have M characters each in next M lines forming MxM grid.

Constraints:

1 <= N <= 10 1 <= M <= 5

output format

Print all words of dictionary present in crossword with space between words.

sample input

2

a aa

2

aa

aa

sample output

a aa

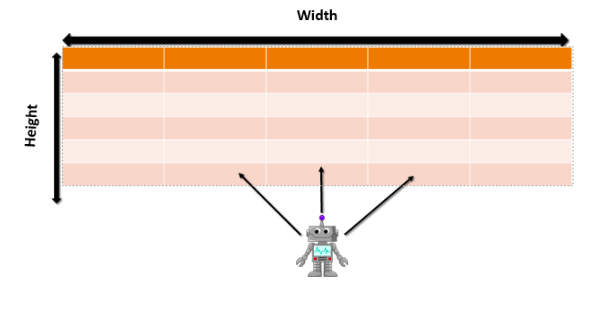
Robot Collect Points

A robot is moving in two dimensional grid. The width of the grid has a fixed value of 5 while the height of the grid will be given. Robot starts to move from the middle of the width and just below the grid. It can move only in three directions.

1.) Left Upward

2.) Upward

3.) Right Upward



Each cell of the grid is filled with a number (-1 or 0 or 1). When the Robot traverse from that cell, he will collect that number and sum up. The Robot has some power to make the value of all the cell of first 5 ∗ 5 subgrids from -1 to 0. He can use that power only once. Can you help the Robot to get the maximum score?

Input Format:

First line contains number t as number of test cases. Each test case contains a integer h as height of grid. Next h lines contains 5 space separated integer which is the value cell of the grid from top to bottom.

Constraints:

• 1 ≤ t ≤ 10 • 1 ≤ h ≤ 12

output format

Print the maximum score collected by the robot durung traversing.

sample input

1

5

0 0 -1 1 0

0 0 0 1 0

0 -1 -1 -1 0

-1 0 1 0 0

0 1 -1 0 0

sample output

4

Rat in a maze

You are given an N\*M grid. Each cell (i,j) in the grid is either blocked, or empty. The rat can move from position (i,j), down or right on the grid.  
Initially rat is on the position (1,1). It wants to reach position (N,M). Find the rightmost path through which, rat can reach this position. A path is rightmost, if the rat is at position (i,j), it will always move to (i,j+1), if there exists a path from (i,j+1) to (N,M).

Input Format:

First line contains 2 integers, N and M, denoting the rows and columns in the grid. Next N line contains. M characters each. An 'X' in position (i,j) denotes that the cell is blocked and ans 'O' denotes that the cell is empty.

Constraints:

1<=N,M<=1000 GRID(i,j)='X' or 'O'

output format

If a solution exists: Print N lines, containing M integers each. A 1 at a position (i,j) denotes that the (i,j)th cell is covered in the path and a 0 denotes that the cell is not covered in the path.  
If solution doesn't exist, just print "-1".

sample input

5 4

OXOO

OOOX

OOXO

XOOO

XXOO

sample output

1 0 0 0

1 1 0 0

0 1 0 0

0 1 1 1

0 0 0 1

Guess Your Way Out

Amr bought a new video game "Guess Your Way Out!". The goal of the game is to find an exit from the maze that looks like a **perfect binary tree of height h**.   
The player is initially standing at the root of the tree and the *exit* from the tree is located at some ***leaf node***.

Let's index all the leaf nodes from the left to the right from 1 to 2h. The exit is located at some node n where 1 ≤ n ≤ 2h, the player doesn't know where the exit is so he has to guess his way out!

Amr follows simple algorithm to choose the path. Let's consider infinite command string "LRLRLRLRL…" (consisting of alternating characters 'L' and 'R'). Amr sequentially executes the characters of the string using following rules:

Character 'L' means "go to the left child of the current node";  
Character 'R' means "go to the right child of the current node";  
If the destination node is already visited, Amr skips current command, otherwise he moves to the destination node;  
If Amr skipped two consecutive commands, he goes back to the parent of the current node before executing next command;  
If he reached a leaf node that is not the exit, he returns to the parent of the current node;  
If he reaches an exit, the game is finished.  
Now Amr wonders, if he follows this algorithm, ***how many nodes he is going to visit before reaching the exit?***

Input Format:

The first line consists of number of test cases T. Each test case consists of two integers h and n.

Constraints:

1 <= T <= 50. 1 <= h <= 50. 1 <= n <= 2^h.

output format

The answer for every test case in new line

sample input

5

3 8

2 3

1 2

2 4

3 6

sample output

14

5

2

6

10