Complexity: 1/10

Finding grade of a student

0-40 U

41-50 D

51-60 C

61-70 B

71-80 B+

81-90 A

91-100 S

Anything outside print “Invalid input”

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is an integer N.

Output:

For each test case, there is one line of output i.e. grade of a student with respective to the given question.

Constraints:

1<=T<=100

0<=N<=300

Example

Input:

3

89

32

300

Output:

A

U

Error

Complexity: 1/10

Write a Program to find the factorial of a given number.

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a integer N.

Output:

For each test case, there is one line of output i.e. factorial of a given number.

Constraints:

1<=T<=100

0<=N<=1000

Example

Input:

2

5

3

Output:

120

6

Complexity: 1/10

Fibonacci series

Generate first N terms of Fibonacci series.

Note: save the file/class as “fibonacci” with appropriate extension

Input:

The first line of input is an integer T, denoting number of test cases. The first line of each test case is first N terms of Fibonacci series.

Output:

For each test case, there is one line of output i.e. first N terms of Fibonacci series.

Constraints:

1<=T<=100

5<=N<=100

Example:

Input:

2

5

10

Output:

0 1 1 2 3

0 1 1 2 3 5 8 13 21 34

Complexity: 1/10

Compare Strings

Program to compare if two strings are equal or not

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is two input strings S1, S2 on the same line separated by a single space respectively.

Output:

For each test case, there is one line of output i.e. whether the Strings are equal or Strings are not equal

Constraints:

1<=T<=100

1<S1, S2<=50

Input:

3

codevita codevita

Hackerrank Hackerearth

Codevita codevita

Example

Output:

Strings are equal

Strings are not equal

Strings are not equal

Complexity: 1/10

Palindrome

Write a Program to check whether the given string is Palindrome or not.

Note: save the file/class as “palindrome” with appropriate extension

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a String S.

Output:

For each test case, there is one line of output i.e. whether it is Palindrome or not a palindrome.

Constraints:

1<=T<=100

2<=S<=50

Example

Input:

2

Malayalam

Hello

Output:

Palindrome

Not a Palindrome

Complexity: 2/10

Write a program to swap even and odd bits of a number represented in binary.

Given an unsigned integer, swap its even and odd bits. The 1st and 2nd bit should be swapped, the 3rd and 4th bit should be swapped, and so on. Print the output in decimal equivalent.

For example, 10101010 should be 01010101. 11100010 should be 11010001.

Input Format :

Input consists of a single integer N.

Output Format :

Single integer of bit-swapped input.

Constraints :

1 <= N <= 10000

Input 1 :

170

Output 1 :

85

Complexity – 2/10

1. Stock Price[] is an array containing stock price of Y company for n days starting from day 1 till nth data. An investor can only buy once and sell once. Write a function which will print day of buying and day of selling so that investor could have maximized his profit.

**Testcase:**

**If Stock Price is Stock Price [ 10, 6, 12, 10, 14, 4]**

**Answer will be:**

**Day of buying: 2**

**Day of selling: 5**

Complexity: 2/10

Keypad typing

You are given a string S of alphabet characters and the task is to find its matching decimal representation as on a mobile phone's numeric keypad. Output the decimal representation corresponding to the string. For ex: if you are given “amazon” then its corresponding decimal representation will be 262966.

Input:

The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case consists of a single line containing a string.

Output:

For each test case, print in a new line, the corresponding decimal representation of the given string.

Constraints:

1 <= T <= 100

1 <= length of String <= 100

Example:

Input

2

itvaccoding

vacquiz

Output

48822263464

8227849

Complexity – 2/10

Write a program to determine whether a given number X is a part of the Fibonacci Sequence.

Write a program to determine whether a given number X is a part of the Fibonacci Sequence. [0,1,1,2,3,5...]

Input Format :

Input consists of only one Integer which is X.

Output Format :

Print True if X is present in the Fibonacci series. Print False if X is not present.

Sample IO :

Test Case1:

Input :

8

Output:

True

Complexity: 2/10

Generate Binary Numbers

Given a number N. The task is to generate and print all binary numbers with decimal values from 1 to N.

Input:

The first line of input contains an integer T denoting the number of test cases. There will be a single line for each testcase which contains N.

Output:

Print all binary numbers with decimal values from 1 to N in a single line.

Constraints:

1 = T = 106

1 = N = 106

Example:

Input:

2

2

5

Output:

1 10

1 10 11 100 101

Explanation:

Testcase 1: Binary numbers from 1 to 2 are 1 and 10

Complexity: 2/10

Numbers divisible by 'K' in a range.

Write a program that given three integers A, B and K, prints the number of integers within the range [A - B] that are divisible by K.

For example, for A = 6, B = 11 and K = 2, your function should return 3, because there are three numbers divisible by 2 within the range [6..11], namely 6, 8 and 10.

Write an efficient algorithm for the following assumptions:

A and B are integers within the range [0..2,000,000,000];

K is an integer within the range [1..2,000,000,000];

A <= B.

Input Format :

The first line of input consists of a single Integer A (lower range).

The second line of input consists of a single Integer B (upper range).

The third line of input consists of a single Integer K.

Output Format :

Output consists of a single integer which is the number of integers divisible by K within the range A-B.

Test Case 1:

Input :

5

15

6

Output :

2

Complexity: 2/10

Count total set/one bits

You are given a number N. Find the total count of set bits (number of 1s when the number is represented in binary) form for all numbers from 1 to N(both inclusive).

Input:

The first line of input contains an integer T denoting the number of test cases. T testcases follow. The first line of each test case is N.

Output:

For each testcase, in a new line, print the total count of all bits.

Constraints:

1 = T = 100

1 = N = 103

Example:

Input:

2

4

17

Output:

5

35

Explanation:

Testcase1:

An easy way to look at it is to consider the number, n = 4:

0 0 0 = 0

0 0 1 = 1

0 1 0 = 1

0 1 1 = 2

1 0 0 = 1

Therefore , the total number of bits is 5.

Complexity: 2/10

Print the unique element which is not repeated.

There is an integer array with duplicate elements. Print the unique element which is not repeated. If more than one elements are found unique print the one with the least value.

Input Format :

N - size of the array

'N' integer values

Output Format :

Single integer from the array which is the unique number

Sample IO :

Input :

9

1 1 2 2 3 4 4 5 5

Output :

3

Complexity: 2/10

Compare Strings – Ignore case

Program to compare if two strings are equal or not ignoring the case

Note: save the file/class as “compareStrings” with appropriate extension

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is two input strings S1, S2 on the same line separated by a single space respectively.

Output:

For each test case, there is one line of output i.e. whether the Strings are equal or Strings are not equal

Constraints:

1<=T<=100

1<S1, S2<=50

Input:

3

codevita codevita

Hackerrank Hackerearth

Codevita codeVita

Example

Output:

Strings are equal

Strings are not equal

Strings are equal

Complexity: 2/10

Prime Numbers

Write a Program to print prime numbers between the given ranges (both boundaries inclusive) and also display largest prime number in the series.

Note: save the file/class as “primeNo” with appropriate extension

Input:

The first line of input is an integer T, denoting number of test cases. The first line of each test case is lower and upper limit, L and U which are positive integers, on the same line separated by a single space respectively.

Output:

For each test case, there are two lines of output. First line is prime number lying between lower and upper limit (both boundaries inclusive).Second line is the largest prime number in the series.

Constraints:

1<=T<=100

0<L<U<=1000

Example:

Input:

2

1 10

11 20

Output:

2 3 5 7

7

11 13 17 19

19

Complexity: 2/10

Pattern

Print the following pattern

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a number of rows N.

Output:

For each test case, there is an output which generates the given pattern with the given number of rows.

Constraints:

1<=T<=100

10<=N<=99

Example

Input:

2

3

4

Output:

1

2 3

4 5 6

1

2 3

4 5 6

7 8 9 10

Complexity: 2/10

Armstrong Number

Check whether the given number is an Armstrong number or not

Note: save the file/class as “armstrongNo” with appropriate extension

Note: It should accept number up to 4-digits.

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a number N upto 4 digits.

Output:

For each test case, there is one line of output i.e. whether it is Armstrong number or not an Armstrong number.

Constraints:

1<=T<=100

1<=N<=9999

Example

Input:

2

370

1500

Output:

Armstrong number

Not an Armstrong number

Complexity: 2/10

Trendy Number

Write a Program to check whether the given number is a Trendy number or not

Note: In a 3 digit number if middle number divisible by 3 then it is trendy number.

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a number N with 3 digits.

Output:

For each test case, there is one line of output i.e. whether it is Trendy number or not a Trendy number.

Constraints:

1<=T<=100

100<=N<=999

Example

Input:

2

464

675

Output:

Trendy Number

Not a trendy Number

Complexity – 2/10

Maximum product of two numbers

Given an array with all elements greater than or equal to zero. Return the maximum product of two numbers possible.

Input:

The first line of input contains an integer T denoting the number of test cases.

The first line of each test case is N, size of array. The second line of each test case contains array elements.

Output:

Print the maximum product of two numbers possible.

Constraints:

1 = T = 100

2 = N = 107

0 = A[i] = 104

Example:

Input:

2

5

1 100 42 4 23

3

20 30 40

Output:

4200

1200

Explanation:

Testcase 1: Two maximum numbers are 100 and 42 and their product is 4200.

Complexity: 2/10

LCM and GCD

Find LCM and GCD of two numbers

Input:

The first line of input is an integer T, denoting number of test cases. The first line of each test case is two integers, N1 and N2 which are positive integers, on the same line separated by a single space respectively.

Output:

For each test case, there are two lines of output. First line is LCM of two given integers N1 and N2. Second line is the GCD of two given integers N1 and N2.

Constraints:

1<=T<=100

0<N1, N2<=1000

Input:

1

24 18

Output:

72

6

Complexity: 2/10

Perfect Square

Check whether the given number is perfect square or not.

Note: save the file/class as “perfectSquare” with appropriate extension

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a number N.

Output:

For each test case, there is one line of output i.e. whether it is Perfect square or not a Perfect square.

Constraints:

1<=T<=100

1<=N<=1000

Example

Input:

2

64

11

Output:

Perfect square

Not a perfect square

Complexity: 2/10

Sum Of Digits

Write a Program to find sum of digits

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a integer N which denotes the digits of a number. The second line of each test case is a number with respective digits N.

Output:

For each test case, there is one line of output i.e. sum of digits of a given number.

Constraints:

1<=T<=100

2<=N<=100000

Example

Input:

2

5

12345

3

123

Output:

15

6

Complexity: 2/10

Find the first repeating element in an array of integers.

Given an array of integers, find the first repeating element in it. We need to find the element that occurs more than once and whose index of the first occurrence is smallest.

Input Format :

N - size of the array

'N' integer values

Output Format :

Single integer from the array which is the first-repeating

Input:

7

10 5 3 4 3 5 6

Output:

5

Explanation: In the above both 5 and 3 are repeating but index of 5 is lower than index of 3 even though both the repeats of 3 occur before the second 5.

Complexity: 2/10

Write a program to print star pattern

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a number of rows N.

Output:

For each test case, there is an output which generates the given pattern with the given number of rows.

Constraints:

1<=T<=100

10<=N<=99

Example

Input:

1

3

Output:

\*

\*\*\*

\*\*\*\*\*

Complexity: 2/10

Decode the pattern

Given a pattern as below and an integer n your task is to decode it and print nth row of it.

The pattern follows as :

11

121

1331

14641

Input:

The first line of input is the number of test cases . Then T test cases follow . The first line of each test case is an integer N.

Output:

For each test case print the required nth row of the pattern.

Constraints:

1<=T<=20

1<=N<=20

Example:

Input:

2

2

4

Output:

11

121

11

121

1331

14641

Complexity 3/10

Find the largest possible prime number which is less than the given number

Input

2

11

7

Output

7

5

Complexity: 3/10

Find the Missing Number

You are given a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in list. One of the integers is missing in the list. Write an efficient code to find the missing integer. If no number is missing print -1.

Input Format :

N - size of the array

'N' integer values

Output Format :

Single integer which is missing in the array

Sample IO :

Input :

7

1 2 4 6 3 7 8

Output :

5

Complexity: 3/10

Special Number

Write a Program to check whether the given number is Special number or not.

Note: split no -->Add and multiply the no and add it to get the same number

Example: if the input is 19 ====>>1+9 = 10 and 1\*9=9 =====>> 10+9=19 which is equal to the input so the given number is a special number.

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a number N.

Output:

For each test case, there is one line of output i.e. whether it is Special number or not a special number.

Constraints:

1<=T<=100

10<=N<=99

Example

Input:

2

19

11

Output:

Special Number

Not a Special Number

Complexity: 3/10

Minimum coin denominations

Find the minimum number of coins required to make n cents.

You can use standard American denominations, that is, 1¢, 5¢, 10¢, and 25¢.

For example, given n = 16, return 3 since we can make it with a 10¢, a 5¢, and a 1¢.

Input Format :

First line of the input is the number of test cases followed by n integers in each line

Output Format :

Single integer which is the minimum number of coins.

Constraints :

1 <= N <= 1000

Input:

2

16

30

Output:

3

3

Complexity: 3/10

Find the Minimum difference pair

Given an unsorted array, find the minimum difference between any pair in given array.

Input:

The first line of input contains an integer T denoting the number of test cases.

The first line of each test case is N, the size of array. Second line of the test case is the Array.

Output:

Print the minimum difference between any two pairs.

Constraints:

1 <= T <= 30

1 < N <= 100

1 <= arr[i] <= 100000

Example:

Input:

2

5

2 4 5 7 9

10

87 32 99 75 56 43 21 10 68 49

Output:

1

6

Complexity: 3/10

Write a Program if given number is divisible by 3, print "foo" , if it divisible by 5, print "Bar", if it is divisible by 3 and 5, print "FooBar" and if it is not divisible by 3 and 5,print "None".

Note: save the file/class as “fooBar” with appropriate extension

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is an integer N

Output:

For each test case, there is one line of output i.e. Foo or Bar or FooBar or None.

Constraints:

1<=T<=100

0<=N<=1000

Example

Input:

4

21

25

30

4

Output:

Foo

Bar

FooBar

None

Complexity – 3/10

Given a list of numbers, create an algorithm that arranges them in order to form the largest possible integer. NOTE: The digits cannot be split. Example 79 cannot be split as 9 first and 7 second

Input Format :

N - size of the array

'N' integer values

Output Format :

Single integer

The first line of input is the integer T, denoting number of test cases. The first line of each test case is an integer N

Output:

For each test case, there is one line of output i.e. Foo or Bar or FooBar or None.

Constraints:

1<=T<=100

0<=N<=1000

Input:

4

10 7 76 415

Output:

77641510

Complexity 3/10

Given two binary numbers add the two numbers in binary form without converting them to decimal value.

eg) a = 1010 b = 11001

output 100011

Complexity: 3/10

Add two numbers given in base n format.

Input a = 123 b = 13 n = 4

Output 202

Complexity: 3/10

Find the k-th smallest element

How to find kth smallest element in an unsorted array?

You are given an unsorted array of numbers and k, you need to find the kth smallest number in the array.

One way to solve this problem is to sort the array in ascending order then pick the k-1th element, that would be your

kth smallest number in array because array index starts at zero, but can you do better?

Input Format :

N - size of the array

'N' integer values

k - smallest number corresponding to the index in ascending order.

Output Format :

Single integer from the array which is the unique number

Sample IO :

Input :

6

10 2 5 3 9 4

2

Output :

3

Complexity: 3/10

How to find common elements in three sorted arrays?

Given three arrays sorted in non-decreasing order, print all common elements in these arrays.

Input Format :

N1 - size of the array1

'N1' integer values

N2 - size of the array2

'N2' integer values

N3 - size of the array3

'N3' integer values

Output Format :

Set of integers common to the above arrays.

Input :

6

1 5 10 20 40 80

5

6 7 20 80 100

8

3 4 15 20 30 70 80 120

Output :

20 80

Complexity – 3/10

Rearrange an array such that arr[i] = i

Given an array of elements of length N, ranging from 1 to N. All elements may not be present in the array. If element is not present then there will be -1 present in the array. Rearrange the array such that A[i] = i and if i is not present, display -1 at that place.

File / Class name should be "valueInIndex".

Input Format :

N - size of the array

'N' integer values

Output Format :

Input array in proper arrangement

Input :

6

6 1 9 3 2 4

Output :

-1

1

2

3

4

-1

6

-1

-1

9

Complexity – 3/10

How to reverse an array in place?

You need to write a program which accepts an integer array and your program needs to reverse that array in place, which means you cannot use additional buffer or array, but one or two variables will be fine.

Of course you cannot use any open source library or method to directly solve this problem and you should not print the array in reverse directly. You need to create your own logic

Input Format :

N - size of the array

'N' integer values

Output Format :

Input array in reverse order

Input :

7

10 5 3 4 3 5 6

Output :

6

5

3

4

3

5

10

Complexity: 3/10

Longest consecutive number sequence

Given an unsorted array of integers, find the length of the longest consecutive elements sequence.

For example, given [100, 4, 99, 1, 3, 2, 200], the longest consecutive element sequence is [1, 2, 3, 4].

Print its length: 4.

Your algorithm should run in O(n) complexity.

Input Format :

First line of the input is an integer N which is the size of the array.

Next N lines of input contains a single integer each representing elements of the array arr[]

Output Format :

Single integer which is the length of the longest consecutive elements sequence.

Constraints :

2 <= N <= 100

0 <= arr[i] <= 10000

Input 1 :

7

100

4

99

1

3

2

200

Output 1 :

4

Complexity: 3/10

Square array elements and sort the resulting array.

Given an unsorted list of integers, square the elements and give the output in sorted order.

For example, given [-9, -2, 0, 2, 3], return [0, 4, 4, 9, 81].

Input Format :

First line of the input is an integer N which is the number of elements in the input array.

Next N lines of input each contains an integer representing the elements of the array.

Output Format :

Single line of output consisting of N space separated integers.

Constraints :

1 <= N <= 100

0 <= arr[i] <= 10000

Input 1 :

5

-9

-2

0

2

3

Output 1 :

0 4 4 9 81

Complexity: 3/10

Partition the given array based on a pivot value.

Given a pivot x, and a list lst, partition the list into three parts.

The first part contains all elements in lst that are less than x. The second part contains all elements in lst that are equal to x. The third part contains all elements in lst that are larger than x Ordering within a part can be arbitrary. The algorithm should result in minimum swaps.

For example, given x = 10 and lst = [9, 12, 3, 5, 14, 10, 10], one partition may be [9, 5, 3, 10, 10, 14, 12].

NOTE: Overall sorting and printing will not be the right answer since it does not take minimum swaps.

Input Format :

First line of the input is an integer N which is the number of elements in the input array arr[].

Next N lines of input each contains an integer representing the elements arr[i].

Last line of input is a single integer which corresponds to the pivot value.

Output Format :

Single line consisting of N space separated integers.

Constraints :

1 <= N <= 100

0 <= arr[i] <= 10000

Input 1 :

7

9

12

3

5

14

10

10

10

Output 1 :

9 5 3 10 10 14 12

Complexity: 3/10

Addition of sub-matrix

Given a matrix C of size N x M. You are given position of sub-matrix as X1, Y1 and X2, Y2 inside the matrix. Find the sum of all elements inside that sub-matrix.

Input:

The first line of input contains an integer T denoting the number of test cases. The first line of each test case is n and m,n is the number of rows and m is the number of columns. The second line of each test case contains C[N][M]. The third line contains four value of X1, Y1, X2, Y2. X1, Y1 is the top left cell and X2, Y2 is the bottom right cell.

Output:

Print the sum of all elements inside that sub-matrix.

Constraints:

1 = T = 15

1 = N, M = 103

1 = C[N][M] = 106

1 <= X1, Y1, X2, Y2 <= M

Example:

Input:

2

5 6

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

3 4 4 5

3 3

9 8 7 4 2 1 6 5 3

1 2 3 3

Output:

78

26

Explanation:

Testcase 2: Sum from cell starting at position (1, 2) (1-based indexing) and ending at (3, 3) is 26.

Complexity – 3/10

Given a String of length N capitalize the first letter of each word in the sentence

Input:

The first line contains T denoting the number of testcases. Then follows description of testcases. Each case contains a string containing dots and characters.

Output:

For each test case, output a String in single line containing the words of the given String with first letter as upper case (capitals)

Constraints:

1<=T<=10

1<=Length of String<=2000

Example:

Input:

2

i like this. program very much

….

Output:

I Like This. Program Very Much

….

Complexity – 3/10

Given a String of length N make the vowels as upper case and consonants as lower case letters

Input:

The first line contains T denoting the number of testcases. Then follows description of testcases. Each case contains a string containing dots and characters.

Output:

For each test case, output a String in single line containing the words of the given String with first letter as upper case (capitals)

Constraints:

1<=T<=10

1<=Length of String<=2000

Example:

Input:

2

i like this. program very much 123

What is thiS $

Output:

I lIkE thIs. prOgrAm VEry Much 123

whAt Is thIs $

Complexity – 3/10

Extract the Number from the String

Benny once had trouble finding the numbers in a string. The numbers are distributed in a string across various test cases.

There are various numbers in each test case you need to find the number in each test case. Each test case has various numbers in sequence.

You need to find only those numbers which do not contain 9. For eg, if the string contains "hello this is alpha 5051 and 9475".You will extract 5051 and not 9475. You need only those numbers which are consecutive, and you need to help him find the numbers.

Input:

The first line consists of T test cases and next T lines contain a string.

Length of each string S may be between 1 to 1000.

Output:

For each string output the number stored in that string if various numbers are there print the largest one. If a string has no numbers print -1.

Constraints:

1<=T<=100

1<=|S|<=1000

Example:

Input:

3

This is alpha 5057 and 97

GSLV F11 was launched in December 19, 2018

GSLV is a satellite

Output:

5057 97

11 19 2018

-1

Complexity – 3/10

Valid Parenthesis String

Given a string containing only three types of characters: '(', ')' and '\*', write a function to check whether this string is valid. We define the validity of a string by these rules:

Any left parenthesis '(' must have a corresponding right parenthesis ')'.

Any right parenthesis ')' must have a corresponding left parenthesis '('.

Left parenthesis '(' must go before the corresponding right parenthesis ')'.

'\*' could be treated as a single right parenthesis ')' or a single left parenthesis '(' or an empty string.

An empty string is also valid.

Note:

The string size will be in the range [1, 100].

Example:

Input 1 :

3

()

(\*)

)(\*)

Output 1 :

True

True

False

Complexity – 3/10

Given an absolute pathname that may have . or .. as part of it, return the shortest standardized path.

For example, given

Input:”/usr/bin/../bin/./scripts/../"

Output: "/usr/bin/".

Input: “/usr/local/../../bin”

Output: “/bin”

Complexity 3/10

Greater on right side "You are given an array A of size N. Replace every element with the next greatest element (greatest element on its right side) in the array. Also, if there is no element next to the last element, replace it with -1.

Input:

The first line of input contains an integer T denoting the number of test cases. T testcases follow. Each test-case contains two lines of input. The first line is N, the size of the array. The second line contains N space separated integers.

Output:

For each test-case, print the modified array.

Constraints:

1 <= T <= 50

1 <= N <= 100

1 <= Ai <= 1000

Example:

Input:

2

6

16 17 4 3 5 2

4

2 3 1 9

Output:

17 5 5 5 2 -1

9 9 9 -1

Explanation:

Testcase1: For 16 the greatest element on its right is 17. For 17 it's 5. For 4 it's 5. For 3 it's 5. For 5 it's 2. For 2 it's -1(no element to its right). So the answer is 17 5 5 5 2 -1

Complexity – 3/10

Complexity – 3/10

Sorting Elements of an Array by Frequency

Given an array A[] of integers, sort the array according to frequency of elements. That is elements that have higher frequency come first. If frequencies of two elements are same, then smaller number comes first.

Input:

The first line of input contains an integer T denoting the number of test cases. The description of T test cases follows. The first line of each test case contains a single integer N denoting the size of array. The second line contains N space-separated integers A1, A2, ..., AN denoting the elements of the array.

Output:

For each test-case, in a new line, print each sorted array in a separate line. For each array its numbers should be separated by space.

Constraints:

1 <= T <= 70

30 <= N <= 130

1 <= Ai <= 60

Example:

Input 1:

2

5

5 5 4 6 4

5

9 9 9 2 5

Output 1:

4 4 5 5 6

9 9 9 2 5

Explanation:

Testcase1: The highest frequency here is 2. Both 5 and 4 have that frequency. Now since the frequencies are same then smaller element comes first. So 4 4 comes first then comes 5 5. Finally comes 6.

The output is 4 4 5 5 6.

Complexity : 3/10

Write a program to find frequency of migrating birds sighting

You have been asked to help study the population of birds migrating across the continent. Each type of bird you are interested in will be identified by an integer value. Each time a particular kind of bird is spotted, its id number will be added to your array of sightings. You would like to be able to find out which type of bird is most common given a list of sightings. Your task is to print the type number of that bird and if two or more types of birds are equally common, choose the type with the smallest ID number.

For example, assume your bird sightings are of types arr=[1,1,2,2,3]. There are two each of types 1 and 2 , and one sighting of type 3. Pick the lower of the two types seen twice: type 1.

Input Format

The first line contains an integer denoting n, the number of birds sighted and reported in the array arr.

The second line describes arr as n space-separated integers representing the type numbers of each bird sighted.

Constraints: 5 <= n <= 2\*10^5

It is guaranteed that each type is 1, 2, 3, 4, or 5.

Output Format

Print the type number of the most common bird; if two or more types of birds are equally common, choose the type with the smallest ID number.

Sample Input 0

6

1 4 4 4 5 3

Sample Output 0

4

Explanation 0

The different types of birds occur in the following frequencies:

Type 1: 1 bird

Type 2: 0 birds

Type 3: 1 bird

Type 4: 3 birds

Type 5: 1 bird

The type number that occurs at the highest frequency is type 4, so we print 4 as our answer.

Complexity: 3/10

Find the Maximum money

Given street of houses (a row of houses), each house having some amount of money kept inside; now there is a thief who is going to steal this money but he has a constraint/rule that he cannot steal/rob two adjacent houses. Find the maximum money he can rob.

Input:

The first line of input contains an integer T denoting the number of test cases.

The first line of each test case is N and money.

Output:

Print maximum money he can rob.

Constraints:

1 <= T <= 100

1 <= money <= 100

1 <= N <= 1000

Example:

Input:

2

5 10

2 12

Output:

30

12

Complexity – 3/10

Twice counter

Given an array of n words. Some words are repeated twice, we need count such words.

Note : Save the file / class as “wordTwiceCounter” with appropriate file extension.

Input:

The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case contains an integer n denoting the number of words in the string. The next line contains n space separated words forming the string.

Output:

Print the count of the words which are repeated twice in the string.

Constraints:

1<=T<=105

1<=no of words<=105

1<=length of each word<=105

Example:

Input:

2

10

hate love peace love peace hate love peace love peace

8

Tom Jerry Thomas Tom Jerry Courage Tom Courage

Output:

1

2

Complexity – 3/10

Given two strings s1 and s2, remove those characters from first string which are present in second string. Both the strings are different and contain only lowercase characters.

Input:

The first line of input contains an integer T denoting the number of test cases.

The first line of each test case is s1,s1 is first string.

The second line of each test case contains s2,s2 is second string.

Output:

Print the modified string(s1). For each test case, print the output in a new line.

Constraints:

1 = T = 15

1 = s2 < s1 = 50

Example:

Input:

2

itvaccodingteam

caw

removeccharaterfrom

string

Output:

itvodingtem

emovecchaaefom

Complexity – 3/10

Implement strstr manually

Your task is to implement the function strstr. The function takes two strings as arguments (s,x) and locates the occurrence of the string x in the string s. The function returns and integer denoting the first occurrence of the string x in s.

Input Format:

The first line of input contains an integer T denoting the no of test cases . Then T test cases follow. Each test case has 2 lines of input, first list the search string s and second line the occurrence of string x

Output Format:

For each test case, in a new line, output will be an integer denoting the output string.

Constraints:

1 <= T <= 100

1<= |s|,|x| <= 1000

Example:

Input

2

ItVACForSuccess

Fr

ITVACForsuccess

For

Output

ItVACForSuccess

ITVACSuccess

Explanation:

Testcase 1: Fr is not present in the string ItVACForSuccess as substring.

Testcase 2: For is present as substring in ITVACForsuccess hence removed and rest of the string is printed

Complexity: 3/10

Print the following X pattern

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is an amount N.

Output:

For each test case, display the X pattern

Constraints:

1<=T<=100

1<=N<=10000

Example

Input:

1

5

Output:

X X

X X

X

X X

X X

Complexity: 4/10

Write a program which prints the number of palindromes within a given string

Input:

The first line of input is the integer T, denoting number of test cases. The rest of the lines have the string

Output:

Number of palindromes

Constraints:

1<=T<=100

1<=N<=10000

Example

Input:

1

Malayalam

Output:

6

Explanation: The palindrome strings are

Malayalam, aya,layal,alayala,ala,ala

Note: Mam is not a palindrome in the above since it is not consecutive

Complexity – 3/10

Binary String

Given a binary string S. The task is to count the number of substrings that start and end with 1. For example, if the input string is “00100101”, then there are three substrings “1001”, “100101” and “101”.

Input:

The first line of input contains an integer T denoting the number of test cases. Each test case consist of an integer 'n' denoting the string length and next line is followed by a binary string.

Output:

For each testcase, in a new line, print the number of substring starting and ending with 1 in a separate line.

Constraints:

1 = T = 40

1 = |S| = 1000

Example:

Input:

2

10101

01101

Output:

3

3

Example:

Testcase 1: There are 6 substrings from the given string. They are 101,10101,101

Testcase 2: There 3 substrings from the given string. They are 11, 101, 1101.

Complexity 4/10

Reverse each word in a given string

Given a String of length N reverse each word in it. Words are separated by spaces. Dot character should not be reversed and should retain its position.

Input:

The first line contains T denoting the number of testcases. Then follows description of testcases. Each case contains a string containing dots, spaces and characters.

Output:

For each test case, output a String in single line containing the reversed words of the given String.

Constraints:

1<=T<=10

1<=Length of String<=2000

Example:

Input:

2

i like this program very much

hello

Output:

i ekil siht margorp yrev hcum

olleh

Complexity: 3/10

Write a program to find longest binary gap in a given number.

A binary gap within a positive integer N is any maximal sequence of consecutive zeros that is surrounded by ones at both ends in the binary representation of N.

For example,

Number 9 has binary representation 1001 and contains a binary gap of length 2.

The number 529 has binary representation 1000010001 and contains two binary gaps: one of length 4 and one of length 3. So 4 is the longest binary gap

The number 20 has binary representation 10100 and contains one binary gap of length 1.

The number 15 has binary representation 1111 and has no binary gaps.

The number 32 has binary representation 100000 and has no binary gaps.

For example, given N = 1041 the function should return 5, because N has binary representation 10000010001 and so its longest binary gap is of length 5. Given N = 32 the function should return 0, because N has binary representation '100000' and thus no binary gaps.

Complexity: 3/10

Write a program to convert the given Hexadecimal number to binary number

Given a Hexadecimal number as an input, the task is to convert that number to Binary number.

Example:

**Input:** Hexadecimal=1AC5

**Output:** Binary=0001101011000101

Complexity – 3/10

Given a **NxNxN** grid. Harvey and Mike are playing a game. The rule of the game is simple. Each of them in their turn choses a cell and mark it with their initials. But the catch here is once a cell is marked they can't chose any of the adjacent(sharing a common face not the just the edge) cell to it. Since Harvey is senior he goes first. Both of them play optimally. The player with no move remaining loses the game. If harvey wins the game print “Harvey” otherwise print “Mike”.

**Input :**  
The first line contains integer **T**, denoting number of test cases. Then T test cases follow. The first line of each test case contains an integer N, denoting the dimensions of the grid.

**Output :**  
Print the answer of each test case in a new line.

**Constraints :**  
1 <= T <= 100  
1 <= N <= 1018

**Example:  
Input :**  
2  
2  
7

**Output :**  
Mike  
Harvey

**Explanation:**  
**Testcase 1:** 2\*2\*2 grid they have 8 blocks, if Harvey marks any of the cell the then Mark has only three options left, and once Mike chooses any one of the three, then Harvey has no other option so mike wins.

Complexity 3/10

Find the union intersection of two list and also find except (remove even elements from list1 and odd elements from list2)

Input

List 1: 1,3,4,5,6,8,9

List 2: 1, 5,8,9,2

Union: 1, 3,4,5,6,8,9,2

Intersection: 1,5,8,9

Except: 1, 3, 5,9,8,2

Complexity: 3/10

Given a set of numbers like we want to find sum of weights based on the following conditions

1. 5 if a perfect square

2. 4 if multiple of 4 and divisible by 6

3. 3 if even number

And sort the numbers based on the weight and print it as follows , Should display the numbers based on increasing order. If two numbers have the same weighted score, then the lowest of the two number is displayed first

Input:

4

25,16,24,21

Output:

21,25,24,16,19

Explanation

Weighted score of 25 = 5 (perfect square), 16 = 5+3,24 = 4+3, 21 = 0, 19=0

The increasing order would be 19,21,25,24,16

Complexity 3/10

The whitewalkers are here and the Great War is on. Each and every living being is giving their best to save the world from the dead. Arya fighting with full dedication found that whitewalkers need to be killed with a pattern or else they wont die. So that she can fight till the end of the war. She tried to understand the pattern. She kills every **'Xth'** whitewalker by stabbing them **'N'** times .  
**Whitewalker approaching order (X)    Number of times-stabbing(N)**  
1                                                                1  
2                                                                1  
3                                                                2  
.                                                                 .  
55                                                              5  
.                                                                 .  
98                                                              3  
101                                                             4  
198                                                              4  
This is the hint of the pattern that Arya needs to follow. Help Arya!

**Input:**   
The first line of the input contains an integer **T,** denoting the number of test cases. The description of each testcase follows. Each test case contains a single line with one integer 'X' the Xth whitewalker  .

**Output:**   
For each test case in, a new line printthe number of stabs required to kill the 'Xth' whitewalker.

**Constraints:**   
1 <= T <= 200  
1 <= X <= 200

**Example:**  
**Input:**  
4  
102  
95  
72  
60

**Output:**  
4  
6  
2  
4

**Explanation:  
Testcase 1:** The number of set bits in the given number will be the number of stabs required to kill that whitewalker.

Complexity – 3/10

Given a 9×9 sudoku we have to evaluate it for its correctness. We have to check both the sub matrix correctness and the whole sudoku correctness.

A sudoku is considered correct if

1. All the cells are filled and no cells are empty
2. The 9 3x3s inside the 9x9 matrix have numbers 1-9 each
3. All rows, columns and diagnols of sudoku matrix has numbers from 1-9 only once

Input:

<81 integer numbers are given one of the after with a space in between>

Output:

True (if Sudoku is correct), False if it is incorrect

Complexity 3/10

Write a program to find the dominator element of an array.

An array A consisting of N integers is given. The dominator of array A is the value that occurs in more than half of the elements of A. For example, consider array A such that

A[0] = 3 A[1] = 4 A[2] = 3 A[3] = 2 A[4] = 3 A[5] = -1 A[6] = 3 A[7] = 3

The dominator of A is 3 because it occurs in 5 out of 8 elements of A (namely in those with indices 0, 2, 4, 6 and 7) and 5 is more than a half of 8.

N is an integer within the range [0..100,000]; each element of array A is an integer within the range [1..2,147,483,647].

Input Format :

First line of input consists of a single integer 'N' which is the size of the array.

Next N lines consists of N integers which are part of the array A.

Output Format :

Output consists of a single integer which is the dominator of the given array. If no such element is found print -1.

Test case 1:

Input :

10

2

3

2

2

3

3

2

2

2

7

Output :

2

Complexity – 3/10

String a is the divisor of string b if and only if there exists a positive integer X such that if we write out string a consecutively X times, we get string b.

For example, string ‘abab’ has two divisors –‘ab’ and ‘abab’.Given 2 strings of length N,M find the number of common divisors for them.

Input Size 1<=M, N<=1000000000

Example

INPUT

abcdabcd

abcdabcdabcdabcd

OUTPUT

2

Complexity: 4/10

Enter two strings and check whether any substring present in first string that follows the pattern of second string. Implement wildcard match for \* and backslash without built in functions.

Input

5

“abcd” “a\*cd”

“aaaa” “a\*”

“a\*c” “a\\*c”

“adsd” “ad”

“abcd” “ac\*d”

Output

Yes

Yes

Yes

No

No

Complexity: 4/10

Alternate Prime & Fibonacci of a given number

Given a number print the first N prime numbers and the Fibonacci of the number.

Input: N

Output Format :

A List of numbers where it is alternating between prime and fibonacci

Input:

7

Output:

2 0 3 1 5 1 7 2 11 3 13 5 17 8

Explanation: 2, 3, 5, 7, 11, 13, 17 are the first 7 prime numbers

0,1,1,2,3,5,8 are the first Fibonacci numbers

The output is alternatively being printed

Complexity: 4/10

Pattern

Print the inverted triangle pattern

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is a number of rows N.

Output:

For each test case, there is an output which generates the given pattern with the given number of rows.

Constraints:

1<=T<=100

10<=N<=99

Example

Input:

2

3

4

Output:

6 5 4

3 2

1

10 9 8 7

6 5 4

3 2

1

Complexity – 4/10

Given an array of size N in which initially all the elements are 0(zero).The task is to count the number of 1’s in the array after performing N moves on the array as explained:

In each move (starting from 1 to N) the element at the position of the multiple of the move number is changed from 0 to 1 or 1 to 0.

**Move 1**: Change the element t position at 1, 2, 3,…

**Move 2**: Change the element t position at 2, 4, 6,…

**Move 3**: Change the element t position at 3, 6, 9,…

Count the elements whose value is 1 after performing N moves.

**Example:**

**Input:** N=5

**Output:** 2

**Explanation:**

**Move 1**: {1,1,1,1,1}

**Move 2**: {1,0,1,0,1}

**Move 3**: {1,0,0,0,1}

**Move 4**: {1,0,0,1,1}

**Move 5**: {1,0,0,1,0}

Total numbers of 1’s after 5 moves=2

Complexity – 4/10

Given a two dimensional array of string like

<”luke”, “shaw”>  
<”wayne”, “rooney”>  
<”rooney”, “Ronaldo”>  
<”shaw”, “rooney”>

Where the first string is “child”, second string is “Father”. And given “ronaldo” we have to find his no of grandchildren Here “ronaldo” has 2 grandchildren. So our output should be 2.

Complexity – 4/10

Alternate sorting: Given an array of integers, rearrange the array in such a way that the first element is first maximum and second element is first minimum. Eg.) Input : {1, 2, 3, 4, 5, 6, 7} Output : {7, 1, 6, 2, 5, 3, 4}

Complexity – 4/10

Remove unbalanced parentheses in a given expression. Eg.) Input : ((abc)((de)) Output : ((abc)(de)) Input : (((ab) Output : (ab)

Complexity – 4/10

Write a program to give the following output for the given input Eg 1: Input: a1b10 Output: abbbbbbbbbb Eg: 2: Input: b3c6d15 Output: bbbccccccddddddddddddddd The number varies from 1 to 99.

Complexity – 4/10

Write a program to sort the elements in odd positions in descending order and elements in ascending order Eg 1: Input: 13,2 4,15,12,10,5 Output: 13,2,12,10,5,15,4 Eg 2: Input: 1,2,3,4,5,6,7,8,9 Output: 9,2,7,4,5,6,3,8,1

Complexity – 4/10

1. Design a cricket dashboard using the input string as the runs.

Example:

W - Wide

O - Out

. – Dot

1,2,3,4,6(runs)

P1…..P11=Player 1 to Player 11

**Input:** …222431666

**Output:**

P1 – 7(runs)

P2 – 25(runs)

Strike – P1

Non-Strike-P2

Total-32

Overs-2

Extra-0

Wicket(s)-0

Remaining Wickets(s)-10

**Input:** WWW….23O11..46

**Output:**

P1 – 16(runs)

P2 – 0(runs)

P3 – 1(runs)

Strike – P3

Non-Strike-P1

Total-20

Overs-2

Extra-3

Wicket(s)-1

Remaining Wickets(s)-9

**Note:** Consider player change the strike for every Over (6 balls).Add one run and one ball to wide. Wide run will not consider under the individual player score.Wide runs consider only as Extra. Total score includes all player individual runs and Extra runs.

Complexity: 4/10

Minimum number of denomination for a given amount

Write a Program to input amount from user and print minimum number of notes (Rs. 500, 100, 50, 20, 10, 5, 2, and 1) required for the amount. How to the minimum number of notes required for the given amount in Programming. Program to find minimum number of notes required for the given denomination. Logic to find minimum number of denomination for a given amount in Program.

Input:

The first line of input is the integer T, denoting number of test cases. The first line of each test case is an amount N.

Output:

For each test case, there is an output i.e. minimum number of denomination for a given amount.

Constraints:

1<=T<=100

1<=N<=10000

Example

Input:

1

575

Output:

Total number of notes:

500: 1

100: 0

50: 1

20: 1

10: 0

5: 1

2: 0

1: 0

Complexity – 4/10

Write a Program to convert the given number to its roman numeral.

Conditions :

0 <= X <= 10000

Input Format :

The first line of input consists of the number of tests cases followed by each test case (X) in a separate line

Output Format :

Print the Roman Numeral corresponding to X.

Test Case1:

4

8

9

9000

40

Output :

VIII

IX

MMMMMMMMM

XL

Complexity – 4/10

Time Conversion

Given a time in 12-hour AM/PM format, convert it to military (24-hour) time.

Note: Midnight is 12:00:00AM on a 12-hour clock, and 00:00:00 on a 24-hour clock. Noon is 12:00:00PM on a 12-hour clock, and 12:00:00 on a 24-hour clock.

Input Format

Number of test cases

A single string s containing a time in 12-hour clock format (i.e.: hh:mm:ssAM or hh:mm:ssPM ), where 01<= hh <= 12 and 00 <= mm, ss <= 59.

Constraints

All input times are valid

Output Format

Convert and print the given time in 24-hour format, where 00 <= hh <= 23.

Sample Input 1 :

1

07:05:45PM

Sample Output 1 :

19:05:45

Complexity – 4/10

Write a program to find if a string is the rotated version of another string.

Given two strings A and B, return whether or not A can be shifted some number of times to get B.

For example, if A is abcde and B is cdeab, print true.

Input Format :

First line of the input is string A.

Second line of the input is another string B.

Output Format :

Single line - "True" if it is possible to get string B upon rotating string A by 'n' times otherwise "False".

Constraints :

1 <= |S| <= 100

Input:

2

abcde

cdeab

abc

acb

Output:

True

False

Complexity – 4/10

Program to find second most frequent character in a given string. If more than one character is second most repeated return “No second most frequent character”

Given a string, find the second most frequent character in it. Expected time complexity is O(n) where n is the length of the input string.

Examples:

Input: str = "aabababa";

Output: Second most frequent character is 'b'

Input: str = "itvacttv";

Output: Second most frequent character is 'v'

Input: str = "ggeeksquizzz";

Output: Second most frequent character is 'g'

The output can also be any other character with

Input: str = "abcd";

Output: No Second most frequent character

Complexity 4/10

Geek and Coffee Shop

Geek loves to drink cold coffee after coding for long hours. One fine day a Geek went to his favorite coffee shop in the town and asked for a cup of cold coffee. The shopkeeper told him that he is their lucky customer and had an offer for the Geek. The offer was that for an amount of N they will fill Geek's cup with N units of cold coffee and after consuming initial N units of coffee the shopkeeper will again refill his cup with half the amount of coffee that Geek consumed in previous refill, and will keep on refilling his cup by half of the previous cycle till the amount to refill becomes nil i.e. 0 (Assume Geek can consume infinite amount of coffee and shop also has infinite amount coffee). Now Geek is curious to know that how many units of coffee will Geek get for Mth refill. Being Geek's friend help him out.

Input: The first line of the input contains an integer T, denoting the number of test cases. The T test case follow. The only line of each test case contains two space separated integers N and M respectively.

Output: For each test case output a single integer on a new line denoting the required answer.

Constraints:

1<=T<=104

1<=N<=109

1<=M<=103

Example:

Input:

2

100 4

10 3

Output:

12

2

Explanation:

For the 4th refill geek will get 12 units of the coffee.

1st Fill: Geek will get 100 units of Coffee

2nd Fill: 100/2 = 50 units

3rd Fill: 50/2 = 25 units

4th Fill: 25/2 = 12 units

Complexity 4/10

Calculate minimum stab spanning all line segments.

Given a set of closed intervals, find the smallest set of numbers that covers all the intervals. If there are multiple smallest sets, return any of them.

For example, given the intervals [0, 3], [2, 6], [3, 4], [6, 9], one set of numbers that covers all these intervals is {3, 6}.

Input Format :

First line of the input is an integer N which is the number of elements in the input array.

Next N lines of input each contains 2 space-separated integers representing the start and end points of a line segment.

Output Format :

Single line of output consisting of N space separated integers.

Constraints :

1 <= N <= 100

0 <= start, end <= 10000

Input 1 :

4

0 3

2 6

3 4

6 9

Output 1 :

3 6

Complexity: 4/10

How Many Xs?

Given an integer X within the range of 0 to 9, and given two positive integers as upper and lower bounds respectively, find the number of times X occurs as a digit in an integer within the range, excluding the bounds. Print the frequency of occurrence as output.

Input:

The first line of input is an integer T, denoting the number of test cases. For each test case, there are two lines of input, first consisting of the integer X, whose occurrence has to be counted. Second, the lower and upper bound, L and U which are positive integers, on the same line separated by a single space, respectively.

Output:

For each test case, there is only one line of output, the count of the occurrence of X as a digit in the numbers lying between the lower and upper bound, excluding them.

Constraints:

1<=T<=100

0<=X<=9

0<L<U<=10^5

Example:

Input:

2

3

100 250

0

20 21

Output:

35

0

Explanation:

In the first test case, the occurrence of 3 in the numbers starting from 101 to 249 is counted and comes out to be 35.

Similarly, for all the other test cases, the occurrence of the given number X is printed as output.

Complexity: 4/10

How to find if there is a sub array with sum equal to zero?

Here you are given an array of positive and negative numbers, find if there is a sub-array of continuous elements with 0 sum.

The first line of input is the integer T, denoting number of test cases. The first line of each test case is an amount N.

Output:

For each test case, there is an output true if there is a sub-array of continuous elements is zero else false

Constraints:

1<=T<=100

1<=N<=10000

Input Format :

T – Number of test cases

N - size of the array

'N' integer values

Output Format :

true - if sum of any sub-array of continuous elements is zero

false - otherwise

Input :

2

5

4 2 -3 1 6

3

1 3 5

Output:

true

false

Explanation: (2+(-3)+1 = 0)

Complexity – 4/10

Segregate 0s and 1s in an array

You are given an array of 0s and 1s in random order. Segregate 0s on left side and 1s on right side of the array. Traverse array only once.

File / Class name should be "arrangeOneZero".

Input Format :

N - size of the array

'N' integers(0 or 1)

Output Format :

Input array in left-right arrangement

Input :

10

0 1 0 1 0 0 1 1 1 0

Output :

0 0 0 0 0 1 1 1 1 1

Complexity – 4/10

Rotate Array

Given an unsorted array arr[] of size N, rotate it by D elements (anti-clockwise).

Input:

The first line of the input contains T denoting the number of testcases. First line of eacg test case contains two space separated elements, N denoting the size of the array and an integer D denoting the number size of the rotation. Subsequent line will be the N space separated array elements.

Output:

For each testcase, in a new line, output the rotated array.

Constraints:

1 <= T <= 200

1 <= N <= 107

1 <= D <= N

0 <= arr[i] <= 105

Example:

Input:

2

5 2

1 2 3 4 5

10 3

2 4 6 8 10 12 14 16 18 20

Output:

3 4 5 1 2

8 10 12 14 16 18 20 2 4 6

Explanation :

Testcase 1: 1 2 3 4 5 when rotated by 2 elements, it becomes 3 4 5 1 2

Complexity – 4/10

Write a program to calculate profit from a set of share prices.

An array A consisting of N integers is given. It contains daily prices of a stock share for a period of N consecutive days. If a single share was bought on day P and sold on day Q, where 0 <= P <= Q < N, then the profit of such transaction is equal to A[Q] ? A[P], provided that A[Q] >= A[P]. Otherwise, the transaction brings loss of A[P] ? A[Q].

For example, consider the following array A consisting of six elements such that:

A[0] = 23171 A[1] = 21011 A[2] = 21123 A[3] = 21366 A[4] = 21013 A[5] = 21367

If a share was bought on day 0 and sold on day 2, a loss of 2048 would occur because A[2] ? A[0] = 21123 ? 23171 = ?2048.

If a share was bought on day 4 and sold on day 5, a profit of 354 would occur because A[5] ? A[4] = 21367 ? 21013 = 354.

Maximum possible profit was 356. It would occur if a share was bought on day 1 and sold on day 5.

N is an integer within the range [0..400,000];

each element of array A is an integer within the range [0..200,000].

Input Format :

First line of input consists of a single integer 'N' which is the size of the array.

Next 'N' lines consists of 'N' integers which are part of array A.

Output Format :

Output consists of a single integer which is the maximum profit that the user can earn.

Test Input 1 :

6

23187

92727

231

4563

123131

1311

Output:

122900

Complexity 4/10

Write a program to find the number of semi-prime numbers within a set of ranges.

A semi-prime is a natural number that is the product of two (not necessarily distinct) prime numbers. The first few semi-primes are 4, 6, 9, 10, 14, 15, 21, 22, 25, 26. You are given two non-empty arrays P and Q, each consisting of M integers. These arrays represent queries about the number of semi-primes within specified ranges.

Query K requires you to find the number of semi-primes within the range (P[K], Q[K]), where 1 <= P[K] <= Q[K] <= N.

For example, consider an integer N = 26 and arrays P, Q such that:

P[0] = 1 Q[0] = 26 P[1] = 4 Q[1] = 10 P[2] = 16 Q[2] = 20

The number of semi-primes within each of these ranges is as follows:

(1, 26) is 10, (4, 10) is 4, (16, 20) is 0.

N is an integer within the range [1..50,000];

each element of arrays P, Q is an integer within the range [1..1000]; P[i] <= Q[i].

Input Format :

First line of input consists of a single integer 'N' which is the size of the array.

Next 2\*N lines consists of N integers which are part of array P and N integers which are part of array Q.

Output Format :

Output consists of a single integer which is the maximum profit that the user can earn.

Test case 1:

4

1

12

34

23

12

23

67

71

Output : 4 4 12 16

Complexity – 4/10

Max Length Chain

You are given N pairs of numbers. In every pair, the first number is always smaller than the second number. A pair (c, d) can follow another pair (a, b) if b < c. Chain of pairs can be formed in this fashion. Your task is to complete the function **maxChainLen** which returns an integer denoting the longest chain which can be formed from a given set of pairs.   
  
**Input:**  
The first line of input contains an integer T denoting the no of test cases then T test cases follow .Then T test cases follow . The first line of input contains an integer N denoting the no of pairs . In the next line are 2\*N space separated values denoting N pairs.  
  
**Output:**  
For each test case output will be the length of the longest chain formed.  
  
**Constraints:**  
1<=T<=100  
1<=N<=100  
  
**Example(To be used only for expected output):  
Input**  
2  
5  
5  24 39 60 15 28 27 40 50 90  
2  
5 10 1 11   
  
**Output**  
3  
1  
​  
**Explanation**  
(i) the given pairs are {{5, 24}, {39, 60}, {15, 28}, {27, 40}, {50, 90} },the longest chain that can be formed is of length 3, and the chain is {{5, 24}, {27, 40}, {50, 90}}  
(ii) The max length chain possible is only of length one.

Complexity – 4/10

Hungry Pizza Lovers

Dominos Pizza has hungry customers waiting in queue. Each unique order ,i, is placed by a customer at time x[i], and the order takes y[i] units of time to complete. You have information of all n orders, Find the order in which all customers will receive their pizza and print it. If two or more orders are completed at same time then sort them by non decreasing order number.

**Input:**  
The first line contains a single integer T, denoting the number of testcases.  
The first line of each test case contains a single integer n, denoting the total no orders.  Each of the  subsequent lines contains two space-separated integers describing the respective values of x[i] and y[i] for order i .

**Output:**  
Print n lines containing order numbers describing the sequence in which the customers receive their pizzas. If two or more customers receive their pizzas at the same time, print the smallest order no first.

**Constraints:**  
1<=T<=10  
1<=n<=500  
1<=i<=n  
1<=x[i],y[i]<=10000

**Example:  
Input:**  
1  
5  
4 1  
6 2  
7 6  
8 1  
1 3  
**Output:**  
5  
1  
2  
4  
3

Complexity – 4/10

Given two sorted arrays, merge them such that the elements are not repeated

Eg 1: Input:

8

2,4,5,6,7,9,10,13

10

2,3,4,5,6,7,8,9,11,15

Output:

2,3,4,5,6,7,8,9,10,11,13,15

Complexity – 4/10

Print the given pattern:

Input: N= 3, M=3

Output:

X X X

X 0 X

X X X   
Input: N=4 M=5

Output:

X X X X

X 0 0 X

X 0 0 X

X 0 0 X

X X X X

Input: N=6 M=7

X X X X X X

X 0 0 0 0 X

X 0 X X 0 X

X 0 X X 0 X

X 0 X X 0 X

X 0 0 0 0 X

X X X X X X

Complexity 4/10

Input: With the starting and ending time of work given find the minimum no of workers needed

Start time end time

1230 0130

1200 0100

1600 1700

Output: 2

Complexity – 4/10

Help john to find new friends in social network

Input: 3

Mani 3 ram raj guna

Ram 2 kumar Kishore

Mughil 3 praveen Naveen Ramesh

Output: Raj guna kumar Kishore praveen Naveen Ramesh

Complexity – 4/10

Valid Parenthesis String with [

Given a string containing only three types of characters: ‘[‘,’]’,’{‘,’}’ '(', ')' and '\*', write a function to check whether this string is valid. We define the validity of a string by these rules:

Any left parenthesis '[' must have a corresponding right parenthesis ']'.

Any left parenthesis '{' must have a corresponding right parenthesis '}'.

Any left parenthesis '(' must have a corresponding right parenthesis ')'.

‘]’ before ‘[‘ is not considered a valid balance

‘}’ before ‘{‘ is not considered a valid balance

‘)’ before ‘(‘ is not considered a valid balance

‘[‘ should be outer most followed be ‘{‘ followed by ‘(‘ and cannot be closed in different order

'\*' could be treated as any of the above characters

Note:

The string size will be in the range [1, 100].

Example:

Input 1 :

4

()

({[]})

[\*()}]

{[]}

]}{[

Output 1 :

True

False

True

False

Explanation: 2 is invalid since the order is [ { [ and not ( { [

Explanation 4: Closing cannot be done before opening

Complexity – 4/10

Given a start word, an end word, and a dictionary of valid words, find the shortest transformation sequence from start to end such that only one letter is changed at each step of the sequence, and each transformed word exists in the dictionary. If there is no possible transformation, return null. Each word in the dictionary have the same length as start and end and is lowercase.

For example, given start = "dog", end = "cat", and dictionary = {"dot", "dop", "dat", "cat"}, return ["dog", "dot", "dat", "cat"].

Given start = "dog", end = "cat", and dictionary = {"dot", "tod", "dat", "dar"}, return null as there is no possible transformation from dog to cat.

Complexity – 4/10

Write a program that computes the length of the longest common subsequence (need not be consecutive) of three given strings. For example, given "epidemiologist", "refrigeration", and "supercalifragilisticexpialodocious", it should return 5, since the longest common subsequence is "eieio".

Complexity – 4/10

For a given number N and size S, generate random alphanumeric strings (N) of each size S which will be used as passwords

Input

3

7

Output

Abcd123

Xyzs232

Ascd343

Complexity – 5/10

[Snakes and Ladders](https://en.wikipedia.org/wiki/Snakes_and_Ladders) is a game played on a 10 x 10 board, the goal of which is get from square 1 to square 100. On each turn players will roll a six-sided die and move forward a number of spaces equal to the result. If they land on a square that represents a snake or ladder, they will be transported ahead or behind, respectively, to a new square. You can assume value to be generated based on your preference in the dice

Find the smallest number of turns it takes to play snakes and ladders for one player

For convenience, here are the squares representing snakes and ladders, and their outcomes:

snakes = {16: 6, 48: 26, 49: 11, 56: 53, 62: 19, 64: 60, 87: 24, 93: 73, 95: 75, 98: 78}

ladders = {1: 38, 4: 14, 9: 31, 21: 42, 28: 84, 36: 44, 51: 67, 71: 91, 80: 100}

Input:

N1 – Number of snakes

[Snake1 square source: Snake1 square dest, … SnakeN1 square source: SnakeN1, square dest]

[Ladder1 square source: Ladder1 square dest, … LadderN1 square source: LadderN1, square dest]

Output:

Integer representing number of moves

Constraints:

1 <= N1 <= 100

1 <= N2 <= 100

Complexity – 5/10

Consider a long alley with a N number of doors on one side. All the doors

are closed initially. You move to and fro in the alley changing the states of

the doors as follows: you open a door that is already closed and you close a

door that is already opened. You start at one end go on altering the state of

the doors till you reach the other end and then you come back and start

altering the states of the doors again.

In the first go, you alter the states of doors numbered 1, 2, 3, ... , n.

In the second go, you alter the states of doors numbered 2, 4, 6 ...

In the third go, you alter the states of doors numbered 3, 6, 9 ...

You continue this till the Nth go in which you alter the state of the door

numbered N.

You have to find the number of open doors at the end of the procedure.

Input:

The first line of input contains a single integer T denoting the number of test

cases. Then T test cases follow. Each test case consists of one line. The line

consists of a positive integer N.

Output:

Corresponding to each test case, in a new line, print the number of doors

that will be open at the end of the procedure mentioned above.

Constraints:

1 ≤ T ≤ 100

1 ≤ N ≤ 1012

Example:

Input

No of test cases : 5

Test Input 1 :

372

Test Input 2 :

2

Test Input 3 :

100

Test Input 4 :

235

Test Input 5 :

567

Output

Test Output 1 : 19

Test Output 2 : 1

Test Output 3 : 10

Test Output 4 : 15

Test Output 5 : 23

Complexity – 5/10

War of Two Nations

Sparta and Persia are fighting a war. A messenger is reporting the same to a remote location using the following format

The first line of input (integer) represents the number of days of battle.

The rest of the rows represents each day of battle. Here H represents Horse, E represents elephants, C represents cavalry and A represents archers. Capital letters represent Spartans and small letters represent Persian army. The . represents the number of units. For example H... represents 3 units of horses, E.... represents 4 units of elephants.

X in the unit represents double points (X is equivalent to two dots)

The objective is to calculate who wins the battle. Every line represents one day of battle.

The team which scores the highest points wins.

Points are calculated as follows:

Horse - 5 points

Elephant - 10 points

Archer - 3 points

Cavalry - 1 point

So the points for

H...E...H.XC..A.. means

5\*3 + 10\*3 + 5\*3 + 1\*2 + 3\*2 = 68 points

Input 1:

5

H...E...H...C..A...h..e....h...c....a....$

H.......................h.......................$

H.XE...H.........C..A...h..e....h.Xc....a....$

H...E...H...CXA...h..e.............h...c....a....$

H...E...H.X.C..A.............hXe....h...c....a....$

Output:

Persia Wins

Complexity – 5/10

Given a matrix mat consisting of \* and lowercase English alphabets, the task is to find the

character which has the maximum number of \* around it (including diagonal elements too). If

two characters have same maximum count, print lexicographically smallest character.

Examples:

Input:

mat[][] = {{'b', '\*', '\*', '\*'},

{'\*', '\*', 'c', '\*'},

{'\*', 'a', '\*', '\*'},

{'\*', '\*', '\*', 'd'}}

Output: a

'a', 'b', 'c' and 'd' are surrounded by

'7', '3', '7' and '3' stars respectively.

'a' and 'c' are surrounded by maximum stars

but 'a' is lexicographically smaller than 'c'.

Input:

mat[][] = {{'\*', 'r', '\*', '\*'},

{'m', 'a', 'z', '\*'},

{'l', '\*', 'f', 'k'},

{'\*', '\*', '\*', 'd'} }

Output: f

Complexity: 5/10

Given an array **arr[]** of **N** non-negative integers representing height of blocks at index **i** as **Ai** where the width of each block is 1. Compute how much water can be trapped in between blocks after raining.  
**Structure is like below**:  
|  |  
|\_|  
We can trap 2 units of water in the middle gap.

**Input:**  
The first line of input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows. Each test case contains an integer **N** denoting the size of the array, followed by **N** space separated numbers to be stored in array.

**Output:**  
Output the total unit of water trapped in between the blocks.

**Constraints:**  
1 <= T <= 100  
3 <= N <= 107  
0 <= Ai <= 108

**Example:  
Input:**  
2  
4  
7 4 0 9  
3  
6 9 9

**Output:**  
10  
0

**Explanation:  
Testcase 1:** Water trapped by block of height 4 is 3 units, block of height 0 is 7 units. So, total unit of water trapped is 10 units.

Complexity: 5/10

Given a string str and an array of strings strArr[], the task is to sort the array according to the

alphabetical order defined by str.

Note: str and every string in strArr[] consists of only lower case alphabets.

Examples:

Input: str = “fguecbdavwyxzhijklmnopqrst”,

strArr[] = {“grammar”, “is”, “the”, “best”, “place”, “for”, “learning”}

Output: for grammar best is learning place the

Input: str = “avdfghiwyxzjkecbmnopqrstul”,

strArr[] = {“rainbow”, “consists”, “of”, “colours”}

Output: consists colours of rainbow

Complexity: 5/10

Given an array **arr[]** of integers, the task is to arrange the array elements such that the last digit of an element is equal to first digit of the next element.

Examples:

*Input: arr[] = {123, 321}  
Output: 123 321*

*Input: arr[] = {451, 378, 123, 1254}  
Output: 1254 451 123 378*

Complexity 5/10

 A reputed college admission team has collected details (studentName, courseOpted, cutoffMarks, location) of the students who has opted for specific course in the college. The team has found there are duplicate entries. Help them remove those duplicates in the collection. An entry is considered duplicate only if all the fields are the same

Name: String

CourseOpted: String

CutoffMarks: Float

Location: String

Input:

5

David Smith, ECE, 194.4,Coimbatore

David Smith, ECE, 194.4,Coimbatore

David Smith, ECE, 193.4,Chennai

Carl Brown, CSE, 197.4,Coimbatore

Output

David Smith, ECE, 194.4,Coimbatore

David Smith, ECE, 193.4,Chennai

Carl Brown, CSE, 197.4,Coimbatore

Complexity: 5/10

Given a C style initialization statement find the variable types and values. Only int and char are allowed. If a variable is not initialized it should be marked as “junk”.

Sample input:

int i,j=4,k=-12;char x=’a’,y=’b’;int c=0;

output:

Integers

i = junk

j = 4

k = -12

c = 0

Characters

x = ‘a’

y = ‘b’

Complexity 5/10

Given bigger NxN matrix and a smaller MxM matrix print TRUE if the smaller matrix can be found in the bigger matrix else print FALSE

Complexity 5/10

You’re given a string as an input. You have to reverse the string by keeping the punctuation and spaces. You have to modify the source string itself with creating an another string. Input :A man, in the boat says : I see 1-2-3 in the sky Output : y kse, ht ni3 21ee slsy : a sta o-b-e ht nin amA

Complexity: 5/10

Electricity Bill Calculation

The consolidated power consumption for a commercial complex is sent in the following format by EB department for a month. The complex has 5 shops each having a separate meter box and the electricity tariff total for a month has to be calculated for each shop separately based on the number of units consumed.

NOTE: The same shop may have more than one entry in a single line. Sometimes if a shop does not consume energy it will not be part of the list. On power shutdown days the entry will be nothing just the date

Find the total power consumed if each shop is charged in the following slabs

First 999 units: 0.40/unit

1000-2000 units: 0.33/unit

2001-5000 units: 0.30/unit

5000+units: 0.20/unit

Input format:

<MMM> <DD>, <YYYY>: <shop 1> <usage 1>$<shop 2> <usage 2>$......<shop n> <usage n>

Input 1:

6

Jan 01, 2020: shop1 320$shop2 220$shop3 330$shop4 420$shop5 57

Jan 02, 2020: shop5 81$shop4 380$shop3 327$shop2 240$shop1 318

Jan 03, 2020: shop1 316$shop3 334$shop4 400$shop5 75$shop2 211

Jan 04, 2020:

Jan 05, 2020: shop1 323$shop2 210$shop3 300$shop4 418$shop5 43

Jan 06, 2020: shop1 324$shop3 315$shop4 411$shop5 48

Output 1:

shop1 bill: 534.93

shop2 bill: 352.40

shop3 bill: 529.98

shop4 bill: 608.70

shop5 bill: 121.60

Complexity: 6/10

Given a matrix of dimension m\*n where each cell in the matrix can have values 0, 1 or 2 which has the following meaning:

0: Empty cell

 1: Cells have fresh oranges

 2: Cells have rotten oranges

So we have to determine what is the minimum time required so that all the oranges become rotten. A rotten orange at index [i,j] can rot other fresh orange at indexes [i-1,j], [i+1,j], [i,j-1], [i,j+1] (up, down, left and right). If it is impossible to rot every orange then simply return -1.

Examples:

Input:  arr[][C] = { {2, 1, 0, 2, 1},

                     {1, 0, 1, 2, 1},

                     {1, 0, 0, 2, 1}};

Output:

All oranges can become rotten in 2 time frames.

Input:  arr[][C] = { {2, 1, 0, 2, 1},

                     {0, 0, 1, 2, 1},

                     {1, 0, 0, 2, 1}};

Output:

All oranges cannot be rotten.

Complexity: 6/10

Given an array **A** of positive integers of size **N**, where each value represents number of chocolates in a packet. Each packet can have variable number of chocolates. There are **M** students, the task is to distribute chocolate packets such that :  
1. Each student gets one packet.  
2. The difference between the number of chocolates given to the students having packet with maximum chocolates and student having packet with minimum chocolates is minimum.

**Input:**  
The first line of input contains an integer **T**, denoting the number of test cases. Then **T** test cases follow. Each test case consists of three lines. The first line of each test case contains an integer **N** denoting the number of packets. Then next line contains **N** space separated values of the array A denoting the values of each packet. The third line of each test case contains an integer m denoting the no of students.

**Output:**  
For each test case in a new line print the minimum difference.

**Constraints:**  
1 <= T <= 100  
1 <=N<= 107  
1 <= Ai <= 1018  
1 <= M <= N

**Example:**  
**Input:**  
2  
8  
3 4 1 9 56 7 9 12  
5  
7  
7 3 2 4 9 12 56  
3

**Output:**  
6  
2

**Explanation:  
Testcase 1:** The minimum difference between maximum chocolates and minimum chocolates is 9-3=6

Complexity 6/10

Complexity 6/10

Write a program to print the number of permutations of strings possible and the strings

Input:

ABC

Output:

6

ABC ACB BAC BCA CBA CAB

Complexity: 6/10

Given a value N, if we want to make change for N cents, and we have infinite supply of each of S = { S1, S2, .. , Sm} valued coins, how many ways can we make the change? The order of coins doesn’t matter.

For example, for N = 4 and S = {1,2,3}, there are four solutions: {1,1,1,1},{1,1,2},{2,2},{1,3}. So output should be 4.

For N = 10 and S = {2, 5, 3, 6}, there are five solutions: {2,2,2,2,2}, {2,2,3,3}, {2,2,6}, {2,3,5} and {5,5}. So the output should be 5.

Complexity 7/10

Given an array of integers A and a sum B, find all unique combinations in A where the sum is equal to B. Each number in A may be used only once in the combination.

Note:  
   All numbers will be positive integers.  
   Elements in a combination (a1, a2, … , ak) must be in non-descending order. (ie, a1 ≤ a2 ≤ … ≤ ak).  
   The combinations themselves must be sorted in ascending order.  
   If there is no combination possible the print "Empty" (without qoutes).  
Example, Given A = 10,1,2,7,6,1,5 and B(sum) 8,

A solution set is:

[1, 7]  
[1, 2, 5]  
[2, 6]  
[1, 1, 6]

**Input:**  
First is T , no of test cases. 1<=T<=500  
Every test case has three lines.  
First line is N, size of array. 1<=N<=12  
Second line contains N space seperated integers(x). 1<=x<=9.  
Third line is the sum B. 1<=B<=30.  
**Output:**  
One line per test case, every subset enclosed in () and in every set intergers should be space seperated.(See example)

**Example:**  
Input:  
2  
7  
10 1 2 7 6 1 5  
8  
5  
8 1 8 6 8  
12

Output:  
(1 1 6)(1 2 5)(1 7)(2 6)

Complexity: 7/10

Pogo Jump 1: You are in an infinite 2D grid where you can move in any of the 8 directions:

(x,y) to (x+1, y), (x - 1, y), (x, y+1), (x, y-1), (x-1, y-1), (x+1,y+1), (x-1,y+1), (x+1,y-1)

You are given a sequence of points and the order in which you need to cover the points. Give the minimum number of steps in which you can achieve it. You start from the first point.

Example:

Input: [(0, 0), (1, 1), (1, 2)]

Output: 2

Move from (0, 0) to (1, 1) in 1 step(diagonal) and then from (1, 1) to (1, 2) in 1 step (rightwards)

Input Format :

First line of the input is an integer N which is the number of points.

Next N lines of input contains 2 integers each separated by a space representing x and y of each point.

Output Format :

Single integer which is the length of the longest consecutive elements sequence.

Constraints :

1 <= N <= 100

0 <= x,y

Input 1 :

3

0 0

1 1

1 2

Output 1 :

2

Complexity: 7/10

Pogo Jump 2

Given an integer list where each number represents the number of hops you can make, determine whether you can reach to the last index starting at index 0.

For example, [2, 0, 1, 0] returns True while [1, 1, 0, 1] returns False.

Input Format :

First line of the input is an integer N which is the number of points.

Next N lines of input each contains integer representing the possible distance of jump from that point.

Output Format :

Single line - "True" if able to reach the end of the array / "False" if not able to reach the end of the array.

Constraints :

1 <= N <= 100

0 <= arr[i] <= 10000

Input 1 :

4

2

0

1

0

Output 1 :

True

Input 2 :

4

1

1

0

1

Output 2 :

False

Complexity – 7/10  
Given a chess board, and N knights placed at positions marked by a-h,1-8 for each knight. Print the total number of positions on the chess board which are not attacked by any knight and which is the position which is attacked by any knight. Also print the position a-h,1-8 which is attacked by maximum knights. If more than one position is most attacked, then print -1. The position where the knight is present is considered as an un-attacked position unless that position is attacked by another knight. Note: A Knight attacks 8 positions as shown below

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  | X |  | X |  |  |  |
| 3 |  | X |  |  |  | X |  |  |
| 4 |  |  |  | N |  |  |  |  |
| 5 |  | X |  |  |  | X |  |  |
| 6 |  |  | X |  | X |  |  |  |
| 7 |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |
|  | a | b | c | d | e | f | g | h |

Input

6

B2B6E6F2G3H4

Output:

39

A4

Explanation: Yellow are un-attacked positions. A4 is attacked by both N1 and N2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  | N1 |  |  |  | N6 |  |  |
| 3 |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  | N5 |
| 5 |  |  |  |  |  |  |  |  |
| 6 |  | N2 |  |  | N3 |  | N4 |  |
| 7 |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |
|  | a | b | c | d | e | f | g | h |

Complexity 8/10

Given a string **s** and a dictionary of words **dict,** add spaces in **s** to construct a sentence where each word is a valid dictionary word.

Return all such possible sentences.

For example, given  
s = "snakesandladder",  
dict = ["snake", "snakes", "and", "sand", "ladder"].

A solution is ["snakes and ladder",  
           "snake sand ladder"].

**Input:**  
The first line contains an integer **T,**denoting the number of test cases.  
Every test case has 3 lines.  
The first line contains an integer **N,**number of words in the dictionary.  
The second line contains **N strings** denoting the words of the dictionary.  
The third line contains a string**s**.

**Output:**  
For each test case, print all possible strings in one line. Each string is enclosed in (). See Example.  
If no string possible print "Empty" (without quotes).

**Constraints:**  
1<=T<=100  
1<=N<=20  
1<=Length of each word in dictionary <=15  
1<=Length of s<=1000

Note: Make sure the strings are sorted in your result.

**Exapmle:**  
**Input:**  
1  
5  
lr m lrm hcdar wk  
hcdarlrm

**Output:**  
(hcdar lr m)(hcdar lrm)

Complexity – 8/10

You are given a 2D matrix, a, of dimension MxN and a positive integer R. You have to rotate the matrix R times and print the resultant matrix. Rotation should be in anti-clockwise direction.

Rotation of a 4x5 matrix is represented by the following figure. Note that in one rotation, you have to shift elements by one step only (refer sample tests for more clarity).

a11←a12←a13←a14← a15

 ↓                                        ↑

a21     a22←a23←a24  a25

 ↓        ↓       ↑ ↑

a31     a32→a33→a34    a35

 ↓                                        ↑

a41→a42→a43→a44→ a45

Example:

First line contains three space separated integers, M,N and R, where M is the number of rows, N is the number of columns in matrix, and R is the number of times the matrix has to be rotated.Then M lines follow, where each line contains N space separated positive integers. These M lines represent the matix. In output print rotated matrix.

Sample:

Input:

4   4 1

1   2 3   4

5   6 7   8

9   10 11   12

13   14 15   16

Output:

2   3 4   8

1   7 11   12

5   6 10   16

9   13 14   15

Input:

2  3 2

1   2 3

4   5 6

Output:

3   6 5

2   1 4

Complexity 8/10

Word Puzzle

Given a 2D board of characters and a word, find if the word exists in the grid.

The word can be constructed from letters of sequentially adjacent cell, where "adjacent" cells are those horizontally or vertically neighboring. The same letter cell may not be used more than once.

For example, given the following board:

[

['A','B','C','I'],

['B','I','C','S'],

['C','D','E','E']

]

The above grid can be initialized in the program itself

Input Format :

First is a number T from 1 to 100 representing the number of test cases

Followed by a set of strings in upper case

Output Format :

Output consists of a single line stating "True" - if the board contains the input string or "False" - if it doesn't contain the input string.

Input:

2

ABCCED

ABCB

Output:

True

False

Complexity – 8/10

Consider a string consisting of the characters < and > only. We consider the string to be balanced if each < always appears before (i.e, to the left of) a corresponding > character (they do not need to be adjacent).Moreover, each < and > act as a unique pair of symbols and neither symbol can be considered as part of any other pair of symbols.

To balance a string, we can replace any > character with <>.Given an expression and a maximum number of replacements, can you turn an unbalanced string into a balanced one?

For example, the strins <<>>, <> , and <><> are all balanced, but the strings >>, <<>, and ><>< are unbalanced. The string >> can be balanced in two moves by replacing each > with a <> to make <><>.

**FUNCTION DESCRIPTION:**

Complete the function balancedOrNot in theeditor below. The function must return an array of integers where element[i] contains a 1 if expression[i] is balanced ora 0 if it is not.

**balancedOrNot has the following parameter(s):**

expression[expression[0]...expression[n-1]]: an array of strings to check

maxReplacements[maxreplacements[0],...maxReplacements[n-1]]:an array of integers representing the maximum number of replacements available for each expressions[i]

**Constraints**

* 1 <= n <= 102
* 1 <= length of expression[i] <= 105
* 0 <= maxReplacement[i] <= 105

**Input Format for custom testing:**

Input from stdin will be processes as follows and passed to the function.

The first line contains an integer n, the size of the array expressions.

The next n lines each contain an element expressions[i].

The next line contains an integer n, the size of the array maxReplacements

The next n lines each contain an element maxReplacements[i]

**Sample Case 0**

Sample input 0

2

<>>>

<>>>>

2

2

2

Sample Output 0

1

0

Explanation 0

e process expressions = /"<>>>" ' "<>>>>"/ and maxReplacements = [2,2] like so:

0. For sting <>>> with maxReplacements[0] = 2, it becomes balanced after two replacements: <>>> ---> <><>> ---> <><><>. The string was converted in <= maxReplacements[0] replacements. Store a 1 in index 0 of our return array.

1. For sting <>>>> with maxReplacements[1] = 2, becomes balanced after three replacements: <>>>> ---> <><>>> ---> <><><>> ---> <><><><>. There were not enough replacements available , so store a 0 in index 1 of the return array.

we then return the array [1,0] as our answer.

**Sample case 1**

sample input 1

2

<>

<>><

2

1

0

sample output 1

1

0

**Explanation 1:**

We process expressions = ["<>", "<>><"] and maxReplacements = [1,0] like so:

0. For string <> with maxReplacements[0] = 1, it is already balanced and needs no replacements. Store a 1 in index 0 of the return array.

1.For string <>>< with maxReplacements[1] = 0, the string is not balanced. It's impossible to balance the string because it ends in < and because there are 0 replacements available. Stroe a o in index 1 of the return array.

We then return the array [1,0]as our answer.

Complexity – 9/10 Write a program to calculate monthly phone bill

Your monthly phone bill has just arrived, and it's unexpectedly large.

You decide to verify the amount by recalculating the bill based on your phone call logs and the phone company's charges. The logs are given as a string S consisting of N lines separated by end-of-line characters (ASCII code 10).

Each line describes one phone call using the following format

""hh:mm:ss, nnn-nnn-nnn""

where ""hh:mm: ss"" denotes the duration of the call (in ""hh"" hours, ""mm"" minutes and ,""ss"" seconds) and ""nnn-nnn-nnn"" denotes the 9-digit phone number of the recipient (with no leading zeros).

Each call is billed separately. The billing rules are as follows:

• If the call was shorter than 5 minutes, then you pay 3 cents for every started second of the call (e.g. for duration ""00:01:07"" you pay 67\*3 = 201 cents).

• If the call was at least 5 minutes long, then you pay 150 cents for every started minute of the call (e.g. for duration ""00:05:00"" you pay 5\* 150 = 750 cents and for duration ""00:05:01"" you pay 6 \*150 = 900 cents).

• All calls to the phone number that has the longest total duration of calls are free. In the case of a tie, if more than one phone number shares the longest total duration, the promotion is applied only to the phone number whose numerical value is the smallest among these phone numbers.

Write a function:

class Solution { public int solution(String 3); } that, given a string S describing phone call logs, returns the amount of money you have to pay in cents.

For example, given string S with N = 3 lines:

"00:01:07,400-234-090

00:05:01,701-080-080

00:05:00,400-234-090"

the function should return 900 (the total duration for number 400-234-090 is 6 minutes 7 seconds, and the total duration for number 701-080-080 is 5 minutes 1 second; therefore, the free promotion applies to the former phone number).

Assume that

• N is an integer within the range 1...100];

• every phone number follows the format ""nnn-nnn-nnn"" strictly; there are no leading zeros; • the duration of every call follows the format ""hh:mm:ss"" strictly (00 <= hh <= 99, 00 <= mm, ss <= 59);

• each line follows the format ""hh:mm:ss,nnn-nnn-nnn "" strictly; there are no empty lines and spaces.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

Complexity – 8/10

You have been hired as a software consultant at a chemist. The chemist shop sells various types of compounds and mixture to their customers. They have a research team that put together various formulas for the chemist of sell. The owner of the shop is a bit of stickler for cleanliness and is also afraid of unforeseen reactions when creating the compounds. So,she has created a rule where a new mixing can is to used for creating a new compound (whether the compound is made of the base elements or from another set of pre-made compounds). Your job as a consultant is to determine the minimal number of bowls that are required to make them. Any compound/element that is part of a definition, without its own definition can be assumed to be a base element. base elements don’t need any preparation.

Multiline input. Where first line N specifies the number of compound definitions followed by N definitions. Followed. Followed by integer M specifies the number of compounds to prepare and M compounds to prepare.

Calculate the minimum number of bowls required to prepared the given compounds.

**Test case 1**

**Input**

2

H2O = H + O

NaCL= Na + CL

1

NaCL

**Output**

1

**Test case 2**

**Input**

2

H2O = H + O

NaCL= Na + CL

2

NaCL

NaCL

**Output**

**1**

**Test case 3**

**Input**

4

H2O = H + O

NaCL = Na + CL

H2SO4 = H2O + S03

S03 = S + O

1

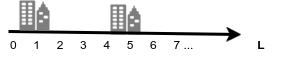
H2SO4

**Output**

3

Complexity: 8/10

On 8th November, 2016, PM of India announced Demonetisation Drive across the country. Use of old currency notes of Rs 500 and Rs 1000 is prohibited and new Rs 2000 and Rs 500 notes are introduced instead. Banks are instructed to fill all the ATMs with new notes. People usually get money from their nearby ATMs but due to unavailability of cash, people have to search for the ATMs having new notes.



In north Delhi, there is a straight road which connects different areas to each other. People usually commute on this road from one area to another. Each area is represented as an integer axis and location of each area is identified with a single coordinate, which is of integer type. No two areas can be on the same coordinate. The distance between two areas will be the absolute value of difference of their coordinates, which is of integer type only. *ATMs having cash to be installed in some area, but not necessarily in all of the areas*. An area and the ATMs in it have the same location. For placing the *new ATM*, their locations must be chosen by the bank, so that the total sum of all distances between each area and its nearest ATM is minimum. **You have to write a program which calculates the least possible sum of all distances between each area and its nearest new ATM.**

**Input Format**

*Argument 1*: It is the integer, which is the number of new ATMs having cash(T) to be installed*.*

*Argument 2:* It contains array of A integers in increasing order. These A integers are the locations of the areas. For each location L, it holds that 1<= L <=10000.

**Constraints**

*1<=T<=30*

*T<= A*

1<= L <=10000

where number of areas is denoted by 'A',  ATMs having cash to be installed is denoted by 'T' and the coordinate of the area is denoted by 'L'.

**Output Format**

You need to return an integer, which is the sum of all distances between each area and its nearest new ATM. Return -1, in case of invalid arguments.

**Sample TestCase 1**

Input

5  
10  
1  
2  
3  
6  
7  
9  
11  
22  
44  
50

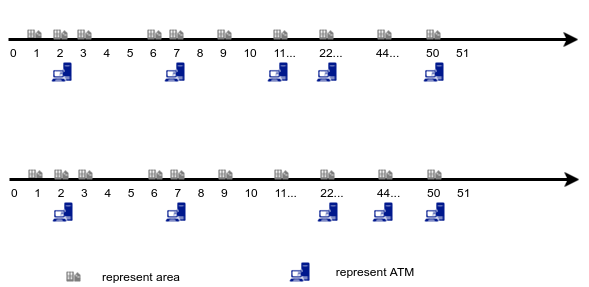
Output

9

Explanation

Number of areas is 10, number of new ATMs to be installed is 5 and locations of 10 areas is {1,2,3,6,7,9,11,22,44,50} as displayed in first image.





If we choose the coordinates 2, 7 , 11, 22 and 50, the sum of all distances between each area and its nearest ATM is 11 and If we choose the coordinates 2, 7 ,22, 44 and 50, sum of all distances between each area and its nearest ATM is 9. **Hence, the least possible sum of all distances between each area and its nearest new ATM will be 9**.

Complexity – 10/10

Year is 3030, water is a scare resource. civilizations live around glaciers in clusters, with a federal body (identified as F) in center melting glaciers and controlling the water distribution. Each cluster has need for water for a day and water for a day and a water storage capacity. Cluster are connected to each other with a pipe identified by \_ . pipes are flow controlled and water flows in forward direction only. Every time water starts flowing through pipe. The clusters drain their tanks for use in other activities, as they can use the water flowing to fill the tanks, and federal body sends water till the capacity is full. Tanks are empty at beginning of day, water tanks fill in an instant. Clusters don’t share their water with other clusters, but allow federal water to flow through the pipe. Federal body releases water at start of day, clusters uses water at end of day. In a pipe link like F-C1--C2-C3-C4. When federal water body targets C3, only C3 and nodes before it (here C1,C2,C3) can fill the tank, C4 can fill it only when its targeted.

Calculate the minimum water needed to help the civilizations survive for n days.

Input is multiple. First line is the number of days to survive. Second line the number of clusters followed by their definitions. Next Is the number of links in the system, followed by the link definitions.

**Custer definition**

**C1 100 300**

**Here**

**C1  - Cluster name**

**100 – daily water need**

**300 – storage capacity**

**Link definition**

**F\_C1**

**F – is the federal source of pipe**

**C – Is the sink destination  of pipe (Cluster tank)**

**Input**

**2**

**3**

**C1 100 300**

**C2 150 300**

**C3 100 100**

**Output**

**1100**

**Input**

**3**

**4**

**C1 100 300**

**C2 100 300**

**C3 100 200**

**C4 100 400**

**4**

**F\_C1**

**F\_C2**

**C2\_C3**

**C3\_C4**

**Output**

**1700**

Complexity – 10/10

Design a blog site that allows you to put in blog posts. Your blog posts are categorized into 5 -sports, travel, science, history and photography.Each week you write blogs in three of the above categories. You are expected to implement the following features.

1.Generate a report of most liked, viewed and commented blog posts.

2. Print the top 5 comments that received the most number of replies for each blog post.

3. Build a suggestion engine to identify which 3 categories need new posts this week.

4. Allow guests to upload attachments in comments and replies for a blog post.

Complexity 10/10

This is about a Prince who wants to rescue her Princess from a bunch of ferocious monsters. You are given a grid of size n+2 and m. Prince initially knows the initial positioning of the monsters on the grid. It is night time so the Prince has an upper hand over the monsters(as the monsters can't see anything at night).

Prince is initially at the top left corner and the Princess is held by the monsters at the bottom right corner. But wait wait wait reaching the other end is not that easy. Prince needs to pass through a forest of size n \* m. You need to get to the other end of the grid but your moves needs to be subtle, you don't want the monsters to know when you pass by(i.e. the position of the monster and Prince can't be same simultaneously).

After every move monster moves to the next position, but take care of the fact that the direction in which monster moves alternates between the rows. Monsters in the first row of the forest moves in left direction, and that of next row in right direction. Once the monster reaches the boundary of the forest it'll teleport to the other end of that row. Prince is allowed to move to any of the adjacent position or it can stay at the same place considering there is no monster at that position as he makes the move.

Help Prince to rescue her Princess from those ferocious monsters.

Note : No monster will be there in the first and last row. Prince can move freely in the first and the last row as there are no monsters there.

Input : There will be two integers n and m, number of rows and columns respectively. The next n+2 lines consists of m characters, this is the initial positioning of the monsters that the Prince knows. The character X indicates a Monster, '.' indicates an empty cell, B indicates the beginning position and T indicates the target position.

Constrains : 1<=n<=50 1<=m<=50

Output : Print the minimum number of moves required to reach the target point. If it is not possible to reach the other end print "Impossible".

Input:

2 3

B..

X..

.X.

..T

Output:

6

Input 2:

10 20

B...................

....XX..XX.XX..XX.XX

..X...XX..X.X.X.X.XX

.....XX.X....X.X..XX

...XXX.....X.XX.XXX.

XXX...X..XXX..X.X.XX

.X..X.X..XX..X...X.X

...XXXX.X..XX....X.X

X..X.XX..XXX.XX..X..

XX..XX..XX.XX..XX...

..X.X...XX..X.XXX..X

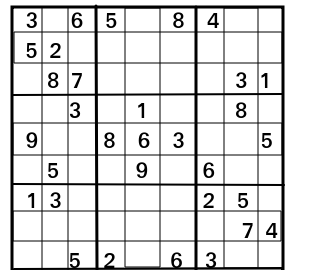
...................T

Output

34

Complexity 10/10

Solve Sudoku:

Given an incomplete Sudoku configuration in terms of a 9 x 9  2-D square matrix (mat[][]). The task to print a solved Sudoku. For simplicity you may assume that there will be only one unique solution.  
  
**Sample Sudoku for you to get the logic for its solution:**  
  


**Input:**  
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains 9\*9 space separated values of the matrix mat[][] representing an incomplete Sudoku state where a 0 represents empty block.

**Output:**  
For each test case, in a new line, print the **space separated values** of the solution of the the sudoku.

**Constraints:**  
1 <= T <= 10  
0 <= mat[] <= 9

**Example:  
Input:**  
1  
3 0 6 5 0 8 4 0 0  
5 2 0 0 0 0 0 0 0  
0 8 7 0 0 0 0 3 1   
0 0 3 0 1 0 0 8 0  
9 0 0 8 6 3 0 0 5  
0 5 0 0 9 0 6 0 0  
1 3 0 0 0 0 2 5 0  
0 0 0 0 0 0 0 7 4  
0 0 5 2 0 6 3 0 0

**Output:**  
3 1 6 5 7 8 4 9 2  
5 2 9 1 3 4 7 6 8  
4 8 7 6 2 9 5 3 1  
2 6 3 4 1 5 9 8 7  
9 7 4 8 6 3 1 2 5  
8 5 1 7 9 2 6 4 3  
1 3 8 9 4 7 2 5 6  
6 9 2 3 5 1 8 7 4  
7 4 5 2 8 6 3 1 9

Long Coding Round – 240 minutes

**BIKE SHOWROOM - REQUIREMENT 1**

**Bike Showroom Requirement - 1**

**Requirement 1:**

Let’s start off by creating two **Bike** objects and check whether they are equal.

Create a **Bike** Class with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private

Create / Generate appropriate Getters & Setters

Add a parameterized constructor to take in all attributes in the given order:   
**Bike(String VIN, String brand, String model,String engineDisplacement, String brakeSystem, Double cost)**

When the “Bike” object is printed, it should display the following details: **[Override the toString method]**  
Print format:  
VIN:"VIN"  
Brand:"brand"  
Model:"model"  
Engine Displacement:"engineDisplacement"  
Brake System:"brakeSystem"  
Cost:"cost"

Two Bikes are considered same if they have the same VIN  and brand(Case – Insensitive). Implement the logic in the appropriate function. **[Override the equals method]**  
  
The input format consists of Bike details separated by comma in the below order,  
**VIN,brand,model,engineDisplacement,brakeSystem,cost**

The Input to your program would be details of two Bike, you need to display their details as given in "5th point(refer above)" and compare the two Bikes and display if the Bikes are same or different.  
  
**Note:**There is an empty line between display statements. Print the empty lines in main function.  
             Display single decimal value for double datatype  
  
**Sample Input & Output 1:**  
  
Enter bike 1 details:  
**MD2GJ3214JR258416,KTM,RC,390cc,Disk,225000.0**  
Enter bike 2 details:  
**MD2GJ3214JR258416,KTM,RC,390cc,Disk,225000.0**  
  
Bike 1  
VIN:MD2GJ3214JR258416  
Brand:KTM  
Model:RC  
Engine Displacement:390cc  
Brake System:Disk  
Cost:225000.0  
  
Bike 2  
VIN:MD2GJ3214JR258416  
Brand:KTM  
Model:RC  
Engine Displacement:390cc  
Brake System:Disk  
Cost:225000.0  
  
Bike 1 is same as Bike 2  
  
**Sample Input & Output 2:**  
  
Enter bike 1 details:  
**MD2GJ3214JR258416,KTM,RC,390cc,Disk,225000.0**  
Enter bike 2 details:  
**WS22UA847JK874510,BMW,S,1000cc,Disk,1800000.0**  
  
Bike 1  
VIN:MD2GJ3214JR258416  
Brand:KTM  
Model:RC  
Engine Displacement:390cc  
Brake System:Disk  
Cost:225000.0  
  
Bike 2  
VIN:WS22UA847JK874510  
Brand:BMW  
Model:S  
Engine Displacement:1000cc  
Brake System:Disk  
Cost:1800000.0  
  
Bike 1 and Bike 2 are different

**BIKE SHOWROOM - REQUIREMENT 2**

**Bike Showroom Requirement - 2**

**Requirement 2:**  
In this requirement, you need to validate the VIN of the bike.  
  
a)Create a class **Main** with the following static methods:

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Boolean validateVIN(String vin) | Validate the VIN based on the rules given below. Returns **true**ifVIN is valid else return **false** |

b) While validating VIN(Vechile Identification Number) follow the below rules. The format of the VIN is given below  
  
Sample format**:1HGBH4G24MN109186**  
  
1.Length of the VIN should seventeen  
2. The first three ,seventh ,tenth and eleventh characters could contains either alphabets(uppercase) or  numeric values.  
3. The fourth and fifth should contains only alphabets(uppercase).  
4. Rest of the characters should contains only numeric values.  
  
**Note:** Print "**VIN is valid**" if refId is valid else print **"VIN is invalid"**.  
            All the above print statements are present in the main method.  
  
**[All text in bold corresponds to input]**  
**Sample Input and Output 1:**  
  
Enter the VIN to be validated:  
**ME1CF2384HN874621**  
VIN is valid  
  
**Sample Input and Output 2:**  
  
Enter the VIN to be validated:  
**MD24H4341JD471245**  
VIN is invalid

Select a Language to Start Solving



[GET EDITOR](javascript:void(0))

**BIKE SHOWROOM - REQUIREMENT 3**

**Bike Showroom - Requirement 3**

**Requirement 3:**  
  
In this requirement, you should find the city wise count of the showrooms.  
  
a) Create a Class **Showroom** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| id | String |
| name | String |
| email | String |
| brand | String |
| city | String |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order:**Showroom(String id, String name, String email, String brand, String city)**  
  
b) Create the following static methods in the **Showroom**class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Map<String,Integer> cityWiseShowroomCount(List<Showroom> list) | This method accepts a List of Showroom objects. It returns a Map with city name as key and the number of showrooms in the city as value in alphabetical order of city name. |

The showroom details should be given as a comma-separated value in the below order,  
**id, name, email, brand, city (the id of the showroom contains '#')**  
  
**Note:**  
Use **System.out.format("%-15s %s\n","City","Count")** for formatting the output.  
Use **TreeMap**in the **cityWiseShowroomCount**method.  
  
**Sample Input/Output 1:**  
  
Enter the number of showrooms  
**5  
#8741BLR,Jack Suzuki,jacksonsuzuki@yahoo.com,Suzuki,Bangalore  
#4785BLR,Balaji Bajaj,balaji.bajaj@yahhoo.com,Bajaj,Bangalore  
#5478HSRA,Sai Ram Bajaj,srirambajaj@gmail.com,Bajaj,Hosur  
#4123CHN,Kick start Suzuki,kickstartsuzuki@gmail.com,Suzuki,Chennai  
#6541CHN,Benjamin Honda,benjaminhonda@gmail.com,Honda,Chennai**  
City                  Count  
Bangalore       2  
Chennai           2  
Hosur               1

**BIKE SHOWROOM - REQUIREMENT 4**

**Bike Showroom - Requirement 4**

**Requirement 4:**  
  
In this requirement, you need to sort the list of bikes based on brand, and engineDisplacement.  
  
a) Create a Class **Bike** with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: **Bike( String VIN, String brand, String model, String engineDisplacement, String brakeSystem, Double cost )**  
  
b) Create the following static methods in the Bike class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| static Bike createBike(String detail) | This method accepts a String. The bike detail separated by commas is passed as the argument. Split the details and create a bike object and returns it. |

The bike details should be given as a comma-separated value in the below order,  
**VIN, brand, model, engineDisplacement, brakeSystem, cost**  
  
c) The Bike class should implement the **Comparable** interface which sorts the Bike list based on brand. While comparing, all the brand attributes in the list are unique.  
  
d) Create a class **EngineDisplacementComparator** which implements Comparator interface and sort the Bike list based on engineDisplacement. While comparing, all the engineDisplacement attributes in the list are unique.  
  
Get the number of Bikes and bike details and create a bike list. Sort the Bikes according to the given option and display the list.  
  
When the “bike” object is printed, it should display the following details  
Print format:  
**System.out.format("%-20s %-10s %-10s %-20s %-12s %s\n","VIN","Brand","Model","Engine Displacement","Brake System","Cost");**  
Display one digit after decimal point for Double datatype.  
  
**Sample Input and Output 1:**

Enter the number of the bikes:  
**3  
MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0  
ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
Enter a type to sort:  
1.Sort by Brand  
2.Sort by Engine Displacement  
**1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD2NH43DLJD471245 | Bajaj | Pulsar | 150cc | Disk | 88000.0 |
| MD6NF25FXHA546812 | TVS | Star City | 110cc | Drum | 49000.0 |
| ME1CF23XBHN874621 | Yamaha | Ray | 120cc | Drum | 55000.0 |

**Sample Input and Output 2:**

Enter the number of the bikes:  
**3  
MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0  
ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
Enter a type to sort:  
1.Sort by Brand  
2.Sort by Engine Displacement  
**2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD6NF25FXHA546812 | TVS | Star City | 110cc | Drum | 49000.0 |
| ME1CF23XBHN874621 | Yamaha | Ray | 120cc | Drum | 55000.0 |
| MD2NH43DLJD471245 | Bajaj | Pulsar | 150cc | Disk | 88000.0 |

**BIKE SHOWROOM - REQUIREMENT 5**

**Bike Showroom - Requirement 5**

**Requirement 5:**  
In this requirement develop a feature in which you can search a List of Bikes by engineDisplacement, and cost.  
  
a) Create a Class Bike with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: Bike( String VIN, String brand, String model, String engineDisplacement, String brakeSystem, Double cost )  
  
b) Create a class **BikeBO**with the following methods,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public List<Bike> findBike(List<Bike> bikeList,String engineDisplacement) | This method accepts a list of bikes and engineDisplacement as arguments and returns a list of bikes that matches with the given engineDisplacement. |
| public List<Bike> findBike(List<Bike> bikeList,Double cost) | This method accepts a list of bikes and cost as arguments and returns a list of bikes that has cost less than the given cost. |

The bike details should be given as a comma-separated value in the below order,  
VIN, brand, model, engineDisplacement, brakeSystem, cost  
  
Print format:  
**System.out.format("%-20s %-10s %-10s %-20s %-12s %s\n","VIN","Brand","Model","Engine Displacement","Brake System","Cost");**  
  
**Note:**The bike lists are displayed in the main method.  
            If any other choice is selected, display "**Invalid choice**"  
            If the search detail is not found, display "No such bike is present"  
            Display one digit after the decimal point for Double Datatype.

**Sample Input/Output 1:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**1**  
Enter the engine displacement:  
**150cc**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD2NH43DLJD471245 | Bajaj | Pulsar | 150cc | Disk | 88000.0 |
| MA4NH43BNJV412631 | Suzuki | Gixxer | 150cc | Disk | 90000.0 |

**Sample Input/Output 2:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**2**  
Enter the cost:  
**50000**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VIN | Brand | Model | Engine Displacement | Brake System | Cost |
| MD6NF25FXHA546812 | TVS | Star City | 110cc | Drum | 49000.0 |

   
**Sample Input/Output 3:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**1**  
Enter the engine displacement:  
**200cc**  
No such bike is present  
   
**Sample Input/Output 4:**  
   
Enter the number of bikes:  
**5**  
**MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0**  
**ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0**  
**MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
**MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0**  
**WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
Enter a search type:  
1.By Engine Displacement  
2.By Cost  
**4**  
Invalid choice

**BIKE SHOWROOM - REQUIREMENT 6**

**Bike Showroom - Requirement 6**

**Requirement 6:**  
In this requirement develop a feature in which you can find the best showroom among the given list of showroms.  
  
a) Create a class **Showroom** with the following attributes,

|  |  |
| --- | --- |
| **Member Field Name** | **Data Type** |
| id | String |
| name | String |
| email | String |
| brand | String |
| city | String |
| bikes | List<Bike> |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: **Showroom(String id, String name, String email, String brand, String city, List<Bike> bikes)**  
  
b) Create a class **Bike** with the following attributes,

|  |  |
| --- | --- |
| **Member Field Name** | **Data Type** |
| VIN | String |
| brand | String |
| model | String |
| engineDisplacement | String |
| brakeSystem | String |
| cost | Double |

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a parameterized constructor to take in all attributes in the given order: **Bike( String VIN, String brand, String model, String engineDisplacement, String brakeSystem, Double cost )**  
  
c) Create the following static method in the Showroom class,

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| Showroom bestShowroom(List<Showroom> showroomList) | this method takes showroom details as argument and returns the best showroom based on the below rules. |

d) The showroom with wide variety of bikes ie., unique bikes with different specifications (engineDisplacement) is said to be the best showroom.  
For Example, consider a showroom A which has 4 different bikes with 120cc, 150cc, 150cc, 150cc and showroom B has 3 different bikes with 150cc, 180cc, 210cc, then showroom B is said to be best showroom as it has more number of unique variety of bikes for the customer to select.  
The showroom details should be given as a comma-separated value in the below order,  
id,name,email,brand,city (the id of the showroom contains '#')  
  
The bike details should be given as a comma-separated value in the below order,  
VIN, brand, model, engineDisplacement, brakeSystem, cost  
  
**Note:**All print lines are displayed in the main method.  
  
**Sample Input/Output :**  
  
Enter the number of showroom:  
**4**  
Enter the showroom details:  
**#1010BLR,Aditya Honda,adityahonda@gmail.com,Honda,Bangalore**  
Enter the number of bikes:  
**3  
MD6NF25FXHA546812,TVS,Star City,110cc,Drum,49000.0  
ME1CF23XBHN874621,Yamaha,Ray,120cc,Drum,55000.0  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0**  
Enter the showroom details:  
**#1102HSRA,Sundhram TVS,sundhramtvs@yahoo.com,TVS,Hosur**  
Enter the number of bikes:  
**4  
MD2NH43DLJD471245,Bajaj,Pulsar,150cc,Disk,88000.0  
MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0  
ME4NH42ANJR985641,Honda,CBR,150cc,Disk,125000.0  
MD2GJ32NBJR258416,KTM,RC,390cc,Disk,225000.0**  
Enter the showroom details:  
**#1546CHN,Racer KTM,racer.ktmchennai@yahoo.com,KTM,Chennai**  
Enter the number of bikes:  
**3  
MA4NH43BNJV412631,Suzuki,Gixxer,150cc,Disk,90000.0  
ME4NH42ANJR985641,Honda,CBR,150cc,Disk,125000.0  
MD2GJ32NBJR258416,KTM,RC,390cc,Disk,225000.0**  
Enter the showroom details:  
**#1452MB,PU Bajaj,pu.bajaj@gmail.com,Bajaj,Mumbai**  
Enter the number of bikes:  
**4  
ME4NH42ANJR985641,Honda,CBR,150cc,Disk,125000.0  
MD2GJ32NBJR258416,KTM,RC,390cc,Disk,225000.0  
JKA1KA12NJA645125,Ninja,RC,300cc,Disk,250000.0  
WS22UA84NJK874510,BMW,S,1000cc,Disk,1800000.0**  
The best Showroom is PU Bajaj