



Q1. The Company Hierarchy

You have the organization chart of a company.

- **Managers (Non-Leaf Nodes):** These are employees who manage at least one other person (they have subordinates below them).
- **Workers (Leaf Nodes):** These are the "boots on the ground" employees who do not manage anyone (they are at the bottom of the chart).

Your task is to analyze the chart and count exactly how many Workers (Leaves) and how many Managers (Non-Leaves) are in the company.

Input Format:

First line contains the total number of employees. Second line contains the employee IDs listed level by level.

Output Format:

Print the number of Workers followed by the number of Managers.

Sample Input

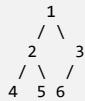
```
6  
1 2 3 4 5 6
```

Sample Output

```
3 3
```

Explanation

The above tree is:



ANSWER:

Test case 1 :

Input:

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

Output:

```
3 3
```

Test case 2 :

Input:

```
0
```

Output:

```
0 0
```

Test case 3 :

Input:

```
1
```

```
2
```

Output:

```
1 0
```

Test case 4 :

Input:

```
2
```

```
1 2
```

Output:

```
1 1
```

Test case 5 :

Input:

```
3
```

```
1 2 3
```

Output:

```
2 1
```

Test case 6 :

Input:

8
1 2 3 4 5 6 7 8

Output:

4 4

Q2. The Shoe Store Inventory

You manage a shoe store where the shoe boxes are organized in a specific structure (a Binary Search Tree) to make searching fast.

A customer comes in and asks for a specific shoe size, K .However, you might not have that exact size in stock. You need to check your inventory system to give the customer two options:

- The "Snug" Fit (Floor): The largest size you have that is smaller than or equal to K.
- The "Loose" Fit (Ceil): The smallest size you have that is larger than or equal to K.

If the customer's exact size (K) exists, both the Snug and Loose options will be K. If a valid option doesn't exist (e.g., the customer wants size 2 but your smallest shoe is size 5), tell them it's unavailable (-1).

Input Format

First line: Number of shoe boxes in stock.

Second line: The sizes of the shoes (arranged in level-order).

Third line: The customer's desired size K.

Output Format

Print the "Snug Fit" size(floor) and "Loose Fit" size(ceil) node values of given key from the binary search tree. If not exists, print -1.

Sample Input

7
4 2 7 1 3 5 8
2

Sample Output

2 2

Explanation

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

For 2, As 2 is itself present in the tree so 2 will be the floor and ceil of itself.

Furthermore, for 6, the largest node value which is smaller than or equal to 6 is 5, and smallest node value which is greater than or equal to 6 is 7. And for 9, the largest node value which is smaller than or equal to 9 is 8, and smallest node value which is greater than or equal to 9 is not in the tree.

ANSWER:

Test case 1 :

Input:

7
4 2 7 1 3 5 8

2

Output:

2 2

Test case 2 :

Input:

0
0

Output:

-1 -1

Test case 3 :

Input:

7
4 2 7 1 3 5 8

6

Output:

5 7

Test case 4 :

Input:

7
4 2 7 1 3 5 8

9
Output:
8 -1

Test case 5 :

Input:
7
8 4 12 2 6 10 14
1
Output:
-1 2

Test case 6 :

Input:
1
5
3
Output:
-1 5

Q3. The Family Reunion

You are organizing a game at a large family reunion. The family structure is recorded as a tree. To keep the game fair, you want to group specific relatives together.

For a chosen family member K, you need to find all their Cousins.

In this game, a "Cousin" is defined strictly as a relative who is at the same generation level (depth) as K, but has a different parent.

Identify and print the IDs of all cousins of K by completing the function **printCousins()** which takes the address of the root node and a key k as a parameter and print the cousins of k separated by space or -1 if no cousin exists.

Input Format

First line contains the total number of nodes, second line contains the node labels separated by space. Third line contains an integer key k whose cousin we have to find.

Output Format

For each test case, print the cousins of given key separated by space in new lines.

Sample Input

6
1 2 3 4 5 6
2

Sample Output

-1

Explanation



The parent of node 2 is 1, who have no siblings, no cousins for node 2.
If key is 6, then 2 is the sibling of parent i.e. 3. So the cousins are 4 and 5.

ANSWER:

Test case 1 :

Input:
6
1 2 3 4 5 6
1
Output:
-1

Test case 2 :

Input:
0
0
0
Output:
-1

Test case 3 :

Input:
6
1 2 3 4 5 6
2
Output:
-1

Test case 4 :

Input:
6
1 2 3 4 5 6
4
Output:
6

Test case 5 :

Input:
6
1 2 3 4 5 6
6
Output:
4 5

Test case 6 :

Input:
6
1 2 3 4 5 6
5
Output:
6

Q4. Brand Logo of a company

A couple decided to open a footwear company, and for its brand logo they wanted to use a single letter. So, they thought of a very interesting method, i.e. any letter from the wife's name that is present at the minimum index in the husband's name will be selected as their company's logo, and if no letter of wife's name is present in husband's name then they will choose letter 'h'.

Now, given the name of both husband and wife, you have to find the logo for their company as per the above method. For example, given husband's name = "nimish" and wife's name = "seema", then two characters (s and m) appears in husband's name, but we will choose 'm' as their company's logo because it appears at minimum index.

Input Format

The First line of input contains an integer T, denoting the no of test cases.
The first line of each test case contains a string, denoting husband name.
The second line of each test case contains a string, denoting wife name.

Output Format

For each test case, print the selected brand logo for their company.

Constraints

1 <= T <= 10
1 <= length_of(name) <= 10^5
All the names consist of lowercase English letters only.

Sample Input

```
2 // Test Cases
nimish
seema
naveen
priti
```

Sample Output

```
m
h
```

ANSWER:

Test case 1 :

Input:
2
nimish
seema
naveen

priti
Output:
m
h

Test case 2 :

Input:
2
codequotient
code
codequotient
t
Output:
c
t

Test case 3 :

Input:
1
codequotient
tne
Output:
e

Test case 4 :

Input:
3
codequotient
gap
codpquoaint
gap
pacific
nihf
Output:
h
p
h

Test case 5 :

Input:
1
codequotient
oent
Output:
o

Test case 6 :

Input:
1
codequotient
qoic
Output:
c

Q5. Decode Enemy Message

Turing intercepts an encoded message from the enemy country, and he has cracked the logic to decode that. The decoding logic is as follows:

The encoded string is composed of multiple words which are separated by space, and to decode that, one needs to reverse the words in the string individually, not the whole string.

Turing needs your help in writing an algorithm to decode those intercepted messages.

Complete the function **decodeMessage()** which takes the string as parameter and reverse the individual words of string and returns it.

Input Format:

The first line of input contains an integer N denoting the no. of messages intercepted.
Next N lines contain a message string each.

Output Format:

For each message string, print its decoded version in a new line.

Sample Input

```
2
edoC tneitouQ sevoL edoC
olleH sredoC
```

Sample Output

```
Code Quotient Loves Code
Hello Coders
```

ANSWER:

Test case 1 :

Input:

2
edoC tneitouQ sevoL edoC
olleH sredoC

Output:

```
Code Quotient Loves Code
Hello Coders
```

Test case 2 :

Input:

6
Welcome to CodeQuotient
a b c d e f

This is a test string

how are you

Coding code program

```
fsqj fsnvlvfdgdjmxvhcv mgzp lbvmfwr wkjbhnhe kcebegiudchncprwrqdljxwie fznwek rzlxrtrdtrojwxlic zqltuxpbqeui hixkqgfniux fhkzwotjqt svnmnkwcfcwdffp  
lxwgimmmxxwprfsqe tvtdy fhulikp qxwzvhxe xwoqducmywckqiy iikrusutnffmkncld zqowuqloy klhgukwkilyfyskboepdxvxqsdlspxoldik mndbnhtzhlyib  
gtlwumbwrdodmvbpocwwsl vukhhiccrvknshzfqcgdfukrvgidhdjnpzbijlkcydttnwrvnepdihdwv fksfvuqpmlukjnlfozkqdjrwwzznfddzfb tebxwbbc  
keuvpienficyjgqsu qrsfdrnd brz wppjqrflkceudnnzh gxzyrzcqjdwxj rfjrgbeqfqtr qipqxbphjcikh yeypnvmysjzh
```

Output:

emocleW ot tneitouQedoC

a b c d e f

sihT si a tset gnirts

woh era uoy

gnidoC edoc margorp

```
jqsf vchvxmjdzgdfvlnf pzgm rwfmvbl ehntbjkw eiwxjldqwrpcnheduiigebek kewnzf cilwxjordtrtxlzc ueqbpxutlqz xuinfqkxih tjtotwzhpf pfddwfewknmnvs  
eqsfrpwxxmmiigwxl ydotvt pkluhf exhvzxwq yviqkewymcudqowx dclcnkmfnntusurkii yolkwwoqz kidlospxldsqxvxdpeobksylikwkuhghlk kbyihztnbdnr  
lswwcopbvmmodmwmwmuwlgt vwdhipdenwrwnttdyckljbzpnjdhjgwrkufdgqczhsnkvrqcihhkv bfzddfnzzwrjdqkzoflnjkulrmpquvskf cbbwxbet  
usqgjycifneipyvuck dnrdfrsqz zrb hnnduecklfrqjppw jxdjzqczryzxg rjtqfqebgrjfr hkcjhpbxqpiq hzsymvpneyd
```

Test case 3 :

Input:

6

It took him a month to finish the meal

Potato wedges probably are not best for relationships

He had a vague sense that trees gave birth to dinosaurs

Toddlers feeding raccoons surprised even the seasoned park ranger

He played the game as if his life depended on it and the truth was that it did

She learned that water bottles are no longer just to hold liquid but they are also status symbols

Output:

tl koot mih a htnom ot hsinif eht laem

otatoP segdew ylbaborp era ton tseb rof spihsnoitaler

eH dah a eugav esnes taht seert evag htrib ot sruasonid

srelddoT gnodeef snooccar desirprus neve eht denosaeas krap regnar

eH deyalp eht emag sa fi sih efil dedneped no ti dna eht hturt saw taht ti did

ehS denrael taht retaw selttob era on regnol tsuj ot dloh diuqil tub yeht era osla sutats slobmys

Test case 4 :

Input:

6

```
Code Quotient Loves Code
Hello Coders
```

h h h

ii jj kk

Hi Hello

```
ijfmhjoedihjeahoyodvyri yoltxfnb vhouaxpmxet wxbrabubtchecofuf aleuffamhoygajyuojswnhjrbumoedhsmwjmgybdaqiuuwrstueocifgbuqdohwpy  
weprdnxqadqjrqcpoqntctislnq sdnxlmhvev jtch zliyfjuifwgwyal bhog yqhgxahtetcmzsttpiucubqzn yzjilfmltwieyln qggghuswfgwesmwqcpba  
nbrsmoevxsbsuxrvebgtecapwmmxxlrdfwawgxmqheufwimiozwgftowjpdzirivu jze al uxdgxhc muyihqvujvxcxapuzqe  
ufmwqzhezvhiurdmfhufwtjhmxovvhjzvlgfyxmolu plbcqqrgyvhjavwsuvdqeq wjovaclz vubxsgoiuiiphgxtp xxgrttwthpjzotodoxoutxxhcrqaciaj
```

Output:

edoC tneitouQ sevoL edoC

olleH sredoC

h h h h

ii jj kk

iH olleH

```
iryvdooyhaejhidleojhmfi bnfxtloy texmpxaurohy fufocehctbhubarbxw ypwodqubfgficoeutsruuiaqadbygmjwmsihdeomubrjhnsjouyjagyoohmaffuel  
qnilikstcntqopcjyqdaqxndrrpew vevhmlxnds hctj laywqgwifuyijlz gohb nzqbcuuptszmngctethaxhqy vnyiyeiwntmfljzy abpcqwmsewgfwssuhhggq  
uvirizdpjwotfgwzoiimiwfuheqmxgawfdrtlxmwwpacetgbevrxxubsxveomrsbn ejz la cxhxgdxu eqzupacxxyjuvwqhyium  
ulomxyfglvwjhzvxomhjtwfuhfmdruivhezechxqwmfu qeqdvvswvajhvygrqqcblp zlcavojw ptxghpiiuoigsxbuv jaicaqrchxxtuoodxotozjphltwtrgxx
```

Test case 5 :

Input:
4
The server crashed during maintenance
Update the software to fix bugs
Install the latest security patch
The database backup was successful
Output:
ehT revres dehsarc gnirud ecananetniam
etadpU eht erawtfos of xif sgub
llatsnI eht tsetal ytiruces htcap
ehT esabatad pukcab saw lufsseccus

Test case 6 :

Input:
6
Configure the network for better performance
User authentication failed multiple times
Cloud storage provides scalable solutions
The system requires a hardware upgrade
Deploy the application on the server
Cybersecurity measures must be updated regularly
Output:
erugifnoC eht krowten rof retteb ecamrofrep
resU noitacitnehtua deliaf elpitum semit
duoIC egarots sedivorp elbalacs snoitulos
ehT metsys seriuer a rawdrh edargpu
yolpeD eht noitacilppa no eht revres
ytirucesrebyC serusaem tsum eb detadpu yrlaluger

Q6. Heena counting characters

Heena loves to count English alphabets from any kind of string, so her mentor has given her a similar kind of task, that is as follows:

Given a string that contains only lowercase English alphabets, she has to print all the distinct alphabets in the sorted order along with their frequencies. For example if the string is “**helloworld**”, then she should print **d1 e1 h1 l3 o2 r1 w1**.

Complete the given function **countFrequency()**, which takes a string as input and prints the frequency of each character as per the above statement.

Input Format

First line contains a string with lowercase characters.

Constraints

'a' <= str[i] <= 'z'
1 <= length of str <= 100000

Output Format

Print all the distinct characters along with their frequency followed by a space. And the characters must be printed in the sorted order.

Sample Input

codequotient

Sample Output

c1 d1 e2 i1 n1 o2 q1 t2 u1

ANSWER:

Test case 1 :

Input:
helloworld
Output:
d1 e1 h1 l3 o2 r1 w1

Test case 2 :

Input:
codequotient
Output:
c1 d1 e2 i1 n1 o2 q1 t2 u1

Test case 3 :

Input:
cccccoodddeeeqqqquuuuoooootttiiieeeeeennnttt
Output:
c4 d3 e7 i4 n3 o7 q4 t8 u4

Test case 4 :

Input:
orpiixzadwfsymraecgxihibutoerogrjpfqnglpcicyvvcumdyhbdbuokzckfnvsnvepgkwxephbhwfjdyinqrptqhmzfyp...
Output:
a3843 b3901 c3931 d4005 e3736 f3841 g3908 h3762 i3806 j3818 k3708 l3768 m3906 n3796 o3939 p3890 q3875 r3839 s3917 t3747 u3771 v3835 w3864 x3898 y3818 z3876

Test case 5 :

Input:
gpwzziupisdwbofnvnezpxzlwtvtnyjcypwzhglsfaemiceufghvgitxlfzlttwumgiveuvjdhtdvvdelahjtzldnzdvuczhzxzizmdudaouvzlzjhwzmbuxzanwutzwicpwthwgnc
Output:
a382 b395 c374 d382 e391 f353 g440 h409 i408 j379 l389 m405 n382 o393 p373 s383 t351 u360 v356 w367 x363 y400 z1565

Test case 6 :

Input:
vrskjipsmmvmpjllpfvomihftfromtmojonorsmftgqmufotlfpoi fpqjitosstlmofrjrrjukhupkuksknvpulfrfshvojunohrvhrqggpoitqpupooikminhvgjpmjukpiggmhphtnoqsi
Output:
f68 g50 h60 i51 j64 k54 l53 m61 n52 o65 p75 q47 r73 s57 t52 u58 v60

Test case 7 :

Input:
msiykccpaekmfkqflfeovossezeryxxrpwwzukpszvsmakvqpqxpopsmcxpzfiuxoawvoyhysfhkcbxxfmveluywakrzmaxzhpwssmcsqazfcirfobeyxzwbahmdwqsheemrc
Output:
a413 b414 c327 d377 e357 f384 h408 i405 k362 l386 m387 o382 p400 q364 r392 s379 u396 v397 w384 x387 y412 z398

Q7. The Word Puzzle Helper

You are building a "cheat tool" for a word puzzle game (like Scrabble). The player has a specific set of letter tiles on their rack.

Your tool needs to show the player every possible way they can arrange their tiles in a line. Even if the arrangement doesn't spell a real word, the player wants to see the combination just in case. The list should be sorted alphabetically to make it easy to read.

Input Format:

First line contains the string of letters on the rack.

Output Format:

Print all permutations of given string in lexicographical order

Sample Input

ABC

Sample Output

ABC
ACB
BAC
BCA
CAB
CBA

ANSWER:

Test case 1 :

Input:
ABC
Output:
ABC
ACB
BAC
BCA
CAB
CBA

Test case 2 :

Input:
code
Output:
cdeo
cdoe
cedo

ceod
code
coed
deco
dcoe
deco
deoc
doce
doec
edo
ecod
edco
edoc
eocd
eodc
ocde
oced
odce
odec
oecd
oedc

Test case 3 :

Input:
hi
Output:
hi
ih

Test case 4 :

Input:
a
Output:
a

Test case 5 :

Input:
CAB
Output:
ABC
ACB
BAC
BCA
CAB
CBA

Test case 6 :

Input:
amit
Output:
aimt
aitm
amit
amti
atim
atmi
iamt
iatm
imat
imta
itam
itma
mait
mati
miat
mita
mtai
mtia
taim
tami
tiam
tima
tmai
tmia

A user is trying to log into an old account but has forgotten exactly how they capitalized their password. They remember the characters and the numbers used (e.g., "pass123"), but they can't remember if it was "Pass123", "paSS123", or "PASS123".

You need to write a recovery script that takes the string the user remembers and generates every possible variation of uppercase and lowercase letters so the system can test them all. Note that numbers and symbols do not have upper/lower case versions and remain unchanged.

Input:

A string s of length n where $1 \leq n \leq 12$. The string contains only alphabetic characters (both uppercase and lowercase).

Output:

Return a list of all possible letter case combinations. The answer may be returned in any order.

Sample Testcase 1:

Input - abc

Output - ["abc", "abC", "aBc", "aBC", "Abc", "AbC", "ABc", "ABC"]

Description -

Sample Testcase 2:

Input - a1b2

Output - ["a1b2", "a1B2", "A1b2", "A1B2"]

ANSWER:

Test case 1 :

Input:

abc

Output:

abc abC aBc aBC Abc AbC ABc ABC

Test case 2 :

Input:

a1c

Output:

a1c a1C A1c A1C

Test case 3 :

Input:

aBc

Output:

abc abC aBc aBC Abc AbC ABc ABC

Test case 4 :

Input:

X3y

Output:

x3y x3Y X3y X3Y

Test case 5 :

Input:

123

Output:

123

Test case 6 :

Input:

A

Output:

a A

Q9. Make a group for competition

There are N students in a class, and their teacher wants to send a group of students for a competition. Assume there is no limit on the number of students that can be part of a group.

Now, given an array of size N denoting their roll no's, determine in how many ways the teacher can create a group out of those N students. For example, If $N = 3$ and $\text{rollNo[]} = \{1, 2, 3\}$, then in 7 of the following ways, the teacher can create a group:

```
[1]
[2]
[3]
[1, 2]
[1, 3]
[2, 3]
[1, 2, 3]
```

Input Format

First line will contain an integer T , denoting the number of test cases.

For each test case:

First line will contain an integer N , denoting the number of students.

Second line will contain N space separated integers, denoting their roll numbers.

Output Format

For each test case, print the total number of ways in which the teacher can create a group, in new lines.

Constraints

```
1 <= T <= 10
1 <= N <= 50
1 <= rollNo[i] <= 10^5
```

Sample Input

```
1      // Test Cases
3      // N (test case 1)
1 2 3 // rollNo[]
```

Sample Output

```
7
```

ANSWER:

Test case 1 :

Input:

```
1
3
1 2 3
```

Output:

```
7
```

Test case 2 :

Input:

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

```
1 2 3 4 5 6 7  
9  
1 2 3 4 5 6 7 8 9  
10  
1 2 3 4 5 6 7 8 9 10  
Output:  
1  
3  
7  
15  
63  
31  
255  
127  
511  
1023
```

Test case 3 :

```
Input:  
3  
14  
1 2 3 4 5 6 7 8 9 10 11 12 13 14  
23  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23  
29  
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
```

```
Output:  
16383  
8388607  
536870911
```

Test case 4 :

```
Input:  
3  
17  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17  
20  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
25  
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
```

```
Output:  
131071  
1048575  
33554431
```

Test case 5 :

```
Input:  
3  
33  
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 31 32 33  
37  
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 31 32 33 34 35 36 37  
40  
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 31 32 33 34 35 36 37 38 39 40
```

```
Output:  
8589934591  
137438953471  
1099511627775
```

Test case 6 :

```
Input:  
3  
42  
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 31 32 33 34 35 36 37 38 39 40 41 42  
45  
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 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45  
50  
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 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

```
Output:  
4398046511103  
35184372088831  
1125899906842623
```