**Inrhythm interview qs**

**Qs:find the bucket where the b no is present and the nos are arranged according to list a.**

**n=int(input())**

**a=list(range(n+1))**

**b=int(input())**

**k=int(input())**

**if b<=n and k<=n//2:**

**print(((index(i)+1)//b)+1 for i in a)**

**else:**

**print("not found")**

a = {1: [1,2,3]}

b = a.copy()

Print(a,b)

a[1].append(4)

Print(a,b)

#### **Deep Copy method.**

It copies all the contents by value.

c = copy.deepcopy(a)

Print(a,b)

a[1].append(5)

Print(a,c)

**def** common\_member(a, b):

    a\_set **=** set(a)

    b\_set **=** set(b)

**if** (a\_set & b\_set):

**print**(a\_set & b\_set)

**else**:

**print**("No common elements")

a **=** [1, 2, 3, 4, 5]

b **=** [5, 6, 7, 8, 9]

common\_member(a, b)

a **=** [1, 2, 3, 4, 5]

b **=** [6, 7, 8, 9]

common\_member(a, b)

**def** common\_member(a, b):

    a\_set **=** set(a)

    b\_set **=** set(b)

    # check length

**if** len(a\_set.intersection(b\_set)) > 0:

**return**(a\_set.intersection(b\_set))

**else**:

**return**("no common elements")

a **=** [1, 2, 3, 4, 5]

b **=** [5, 6, 7, 8, 9]

print(common\_member(a, b))

a **=**[1, 2, 3, 4, 5]

b **=**[6, 7, 8, 9]

print(common\_member(a, b))

a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

c = [x for x in a for y in b if x == y]print (set(c))

**import** java.util.ArrayList;

**public** **class** GFG {

    // main method

**public** **static** **void** main(String[] args)

    {

        // create ArrayList list1

        ArrayList<String>

            list1 = **new** ArrayList<String>();

        // Add values in ArrayList

        list1.add("Hii");

        list1.add("Geeks");

        list1.add("for");

        list1.add("Geeks");

        // print list 1

        System.out.println("List1: "

                           + list1);

        // Create ArrayList list2

        ArrayList<String>

            list2 = **new** ArrayList<String>();

        // Add values in ArrayList

        list2.add("Hii");

        list2.add("Geeks");

        list2.add("Gaurav");

        // print list 2

        System.out.println("List2: "

                           + list2);

        // Find the common elements

        list1.retainAll(list2);

        // print list 1

        System.out.println("Common elements: "

                           + list1);

    }

}

**import sys**

**import re**

**TEST\_RE = re.compile(r'^(\d{1,3})[- ](\d{1,3})[- ](\d{4,10})\n?$')**

**if \_\_name\_\_ == '\_\_main\_\_':**

**sys.stdin.readline()**

**for line in sys.stdin:**

**print(TEST\_RE.sub('CountryCode=\g<1>,LocalAreaCode=\g<2>,Number=\g<3>', line))**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.NoSuchElementException;**

**import java.util.Scanner;**

**public class Solution {**

**public static void main(String[] args) {**

**Scanner in = new Scanner(System.in);**

**List<String> program = new ArrayList<String>();**

**String line;**

**try {**

**while ((line = in.nextLine()) != null) {**

**if (!line.equals("")) {**

**program.add(line.trim());**

**}**

**}**

**} catch (NoSuchElementException e) {**

**}**

**String imp = program.get(0);**

**if ((imp.startsWith("package") || imp.startsWith("import"))**

**&& imp.endsWith(";")) {**

**System.out.println("Java");**

**return;**

**}**

**if (imp.startsWith("#include")) {**

**System.out.println("C");**

**return;**

**}**

**boolean python = true;**

**for (String s : program) {**

**if (python && s.endsWith(";")) {**

**python = false;**

**break;**

**}**

**}**

**if (python) {**

**System.out.println("Python");**

**return;**

**}**

**for (String s : program) {**

**if (s.contains("class ") || s.contains("interface ")) {**

**System.out.println("Java");**

**return;**

**}**

**}**

**System.out.println("C");**

**}**

**}**

**import fileinput**

**stdin = [line.strip() for line in fileinput.input()]**

**tokens = []**

**for line in stdin:**

**tokens.extend(line.split())**

**tokens = set(tokens)**

**if 'class' in tokens and '{' in tokens:**

**print('Java')**

**elif '{' in tokens:**

**print('C')**

**else:**

**print('Python')**

**Rotate**

**Array**

Given an unsorted array **arr[]** of size **N**, rotate it by **D** elements in the counter-clockwise direction.

#User function Template for python3

def rotateArr(A,D,N):

#Function to left rotate arr[] of size n by d

for i in range(gcd(D,N)):

# move i-th values of blocks

temp = A[i]

j = i

while 1:

k = j + D

if k >= N:

k = k - N

if k == i:

break

A[j] = A[k]

j = k

A[j] = temp

**Find Immediate Smaller Than X**

Given an array **arr[]** of size **N** containing positive integers and an integer **X**. You need to find the value in the array which is**smaller than X and closest to it**.

import math

def immediateSmaller(arr,n,x):

#return required ans

for k in range(len(arr)):

if abs(arr[k]-x)>=1 and abs(arr[k]-x)<=len(arr)//2:

return arr[k]

elif abs(arr[k]-x)==0:

return "-1"

#code here

**Leaders in an array**

Given an array A of positive integers. Your task is to find the leaders in the array.

**Note:** An element of array is leader if it is greater than or equal to all the elements to its right side. Also, the rightmost element is always a leader.

def leaders(A,N):

#Your code here

'''

Just return the list with leaders in it

'''

v=[]

for i in range(0,N-1):

if A[i]>j for j in A[i+1:n]:

v.append(A[i])

return v

**Reverse array in groups**

Given an array **arr[]** of positive integers of size **N**. Reverse every sub-array of **K**group elements.

def reverseInGroups(A,N,K):

#Your code here

l=[]

for k in reversed(A[:K]):

l.append(k)

for p in reversed(A[k:]):

l.append(p)

return l

**Count distinct elements in every window**

Given an array of integers and a number K. Find the count of distinct elements in every window of size K in the array.

def countSet(a):

return len(set(a))

def countDistinct(arr, n, k):

l=[]

for i in range(n-k+1):

l.append(countSet(arr[i:i+k]))

return l

or

from collections import defaultdict

def countDistinct(arr, n, k):

# Creates an empty hashmap mp

mp = defaultdict(lambda:0)

# initialize distinct element

# count for current window

dist\_count = 0

# Traverse the first window and store count

# of every element in hash map

for i in range(k):

if mp[arr[i]] == 0:

dist\_count += 1

mp[arr[i]] += 1

res = []

# Print count of first window

res.append (dist\_count)

# Traverse through the remaining array

for i in range(k, n):

# Remove first element of previous window

# If there was only one occurrence,

# then reduce distinct count.

if mp[arr[i - k]] == 1:

dist\_count -= 1

mp[arr[i - k]] -=1

# Add new element of current window

# If this element appears first time,

# increment distinct element count

if mp[arr[i]] == 0:

dist\_count += 1

mp[arr[i]] += 1

# Print count of current window

res.append (dist\_count)

return res

Close

**Find Transition Point**

You are given a sorted array containing only numbers 0 and 1. Find the transition point efficiently. Transition point is a point where "0" ends and "1" begins (0 based indexing).  
Note that, if there is no "1" exists, print -1.

def transitionPoint(arr, n):

#Code here

k="".join(map(str,arr))

k1=k.rindex("0")

if abs(k1-k.index("1"))==1:

return k1

else:

return "-1"

**Equilibrium Point**

Given an array A of N positive numbers. The task is to find the first Equilibium Point in the array.   
Equilibrium Point in an array is a position such that the sum of elements before it is equal to the sum of elements after it.

def equilibriumPoint(A, N):

# Your code here

if len(A)%2!=0:

if len(A)==1:

return "1"

elif sum(A[0:(len(A)//2)-1])==sum(A[(len(A)//2)+1:len(A)-1]):

return str(int(len(A)//2))

else:

return "-1"

else:

return "-1"

**Subarray with given sum**

Given an unsorted array **A**of size **N** of non-negative integers, find a continuous sub-array which adds to a given number **S**.

def subArraySum(arr, n, sum):

# Initialize curr\_sum as

# value of first element

# and starting point as 0

curr\_sum = arr[0]

start = 0

# Add elements one by

# one to curr\_sum and

# if the curr\_sum exceeds

# the sum, then remove

# starting element

i = 1

while i <= n:

# If curr\_sum exceeds

# the sum, then remove

# the starting elements

while curr\_sum > sum and start < i-1:

curr\_sum = curr\_sum - arr[start]

start += 1

# If curr\_sum becomes

# equal to sum, then

# return true

if curr\_sum == sum:

print ("%d %d"%(start+1, i),end=" ")

return 1

# Add this element

# to curr\_sum

if i < n:

curr\_sum = curr\_sum + arr[i]

i += 1

# If we reach here,

# then no subarray

print ("-1",end=" ")

Close

**Maximum of all subarrays of size k**

Given an array **A** and an integer **K**. Find the maximum for each and every contiguous subarray of size K.

#Python 3

def max\_of\_subarrays(arr,n,k):

'''

you can use collections module here.

:param a: given array

:param n: size of array

:param k: value of k

:return: A list of required values

'''

res = []

d = deque()

for i in range(k):

while len(d) and arr[i]>=arr[d[-1]]: # -1 repesents right end of deque

d.pop()

d.append(i)

for i in range(k,n):

# print(arr[d[0]],end=" ") # print element at front of deque

res.append (arr[d[0]])

while len(d) and d[0]<=i-k:

d.popleft()

while len(d) and arr[i]>=arr[d[-1]]:

d.pop()

d.append(i)

res.append (arr[d[0]])

d.popleft()

return res

Close

**Longest Subarray Of Evens And Odds**

You are given an array of size n. Find the**maximum possible length**of a subarray such that its elements are arranged alternately either as**even and odd** or**odd and even** .

def maxEvenOdd(arr,n):

msf=1

cmax=1

for i in range(1,n):

if ((arr[i-1]%2==0 and arr[i]%2 !=0) or (arr[i-1]%2 !=0 and arr[i]%2==0)):

cmax+=1

msf=max(msf,cmax)

else:

cmax=1

return msf

**Maximum Subarray Sum**

Given an array **arr[]**of **N** integers. Find the contiguous sub-array with maximum sum.

def maxSubArraySum(array):

'''

Returns the maximum subarray sum.

array (list): A list of numbers.

'''

n = len(array)

max\_sum = array[0]

curr\_sum = array[0]

for i in range(1, n):

if curr\_sum < 0:

curr\_sum = 0

curr\_sum += array[i]

max\_sum = max(max\_sum, curr\_sum)

return max\_sum

Close

**Binary Search**

Given a sorted array **A[]**(0 based index) and a key**"k"**  you need to complete the function **bin\_search**to  determine the position of the key if the key is present in the array. If the key is not  present then you have to return **-1**. The arguments left and right denotes the left most index  and right most index of the array **A[]**. There are multiple test cases. For each test case, this function will be called individually.

from bisect import \*

def bin\_search(arr, left, high, key):

#Code here

high-=1

i = bisect(arr, key, left, high)

if i == 0:

return (-1)

elif arr[i-1] == x:

return (i-1)

else:

return (-1)

or

class GfG

{

int bin\_search(int A[],int left,int right, int k)

{

if(left>right)

return -1;

int mid = (left+right)/2;

if(A[mid]==k)

return mid;

if(A[mid]>k)

return bin\_search(A,left,mid-1,k);

else

return bin\_search(A,mid+1,right,k);

}

}

**Remove duplicate elements from sorted Array**

Given a sorted array **A**of size **N**. The function **remove\_duplicate**takes two arguments . The first argument is the sorted array **A[ ]** and the second argument is**'N'**the size of the array and returns the size of the new converted array **A[ ]**with no duplicate element.

**Maximum Difference**

Given array A[] of integers, the task is to complete the function **findMaxDiff** which finds the maximum absolute difference between nearest left and right smaller element of every element in array.If the element is the leftmost element, nearest smaller element on left side is considered as 0. Similarly if the element is the rightmost elements, smaller element on right side is considered as 0.

**Smallest Positive missing number**

You are given an array **arr**[] of **N** integers including 0. The task is to find the smallest positive number missing from the array.  
**Note:**Expected solution in O(n) time using constant extra space (**don't**use **hash maps** or **sorting**algorithms in your solution).

def missingNumber(arr,n):#Our original array

m = max(arr) #Storing maximum value

if m < 1:

# In case all values in our array are negative

return 1

if n == 1:

#If it contains only one element

return 2 if arr[0] == 1 else 1

l = [0] \* m

for i in range(n):

if arr[i] > 0:

if l[arr[i] - 1] != 1:

#Changing the value status at the index of our list

l[arr[i] - 1] = 1

for i in range(len(l)):

#Encountering first 0, i.e, the element with least value

if l[i] == 0:

return i+1

#In case all values are filled between 1 and m

return i+2

Close

* [](https://www.geeksforgeeks.org/)
* 5th
* **First Repeating Element**
* Given an array **arr**[] of size **N**. The task is to find the first repeating element in an array of integers, i.e., an element that occurs more than once and whose index of first occurrence is smallest.
* def firstRepeated(arr, n):
* ans=9999999999999
* d=dict()
* for i,e in enumerate(arr):
* if e in d:
* if d[e]<ans and ans!=-1:
* ans=d[e]
* else:
* d[e]=i
* if ans!=9999999999999:
* return(ans+1)
* else:
* return(-1)
* **Peak element**
* Given an array **A** of **N** integers. The task is to find a **peak element in it in O( log N ) .**  
  An array element is peak if it is not smaller than its neighbours. For corner elements, we need to consider only one neighbour.  
  **Note:** There may be multiple peak element possible, in that case you may **return any valid index (0 based indexing)**.
* def peakElement(arr, n):
* # Code here
* if n is 1:
* return 0
* for i in range(n):
* # if element at first index is greater than next
* if i==0 and arr[1]<arr[0]:
* return 0
* # if element is at last index and it is greater than
* # its prev one
* elif i==n-1 and arr[n-2]<arr[n-1]:
* return n-1
* # case, when element is at any other index
* # then you need to check both of its neighbour
* elif arr[i-1]<arr[i] and arr[i]>arr[i+1]:
* return i
* Close
* **Minimum Swaps to Sort**
* Given an array of integers. Find the minimum number of swaps required to sort the array in non-decreasing order.

def minSwaps(a,n):

arrPos=[]

for i in range(n):

arrPos.append([a[i],i])

arrPos.sort()

vis=[0 for i in range(n)]

ans=0

for i in range(n):

if(vis[i] or arrPos[i][1]==i):

continue

cycle\_size=0

j=i

while(not vis[j]):

vis[j]=1

j=arrPos[j][1]

cycle\_size+=1

ans+=cycle\_size-1

return ans

**Union of Two Sorted Arrays**

Given two sorted arrays **arr**[] and **brr**[] of size **N** and **M** respectively. The task is to find the union of these two arrays.  
Union of two arrays can be defined as the common and distinct elements in the two arrays.

def mergeArrays(a,b,n,m):

'''

:param a: given sorted array a

:param n: size of sorted array a

:param b: given sorted array b

:param m: size of sorted array b

:return: The union of both arrays as a list

'''

# code here

d=sorted(set(a+b))

d1="".join(map(str,d))

return d1

**Intersection of two sorted arrays**

Given two sorted arrays **arr1**[] and **arr**2[] of sizes **N** and **M** respectively. The task is to find **intersection**of the two arrays.  
Intersection of two arrays contains the elements common to both the arrays. The intersection should not count duplicate elements

def printIntersection(arr1,arr2,n,m):

i=0

j=0

flag=False

l = []

while(i<n and j<m):

if i>0 and arr1[i-1] == arr1[i]:

i+=1

continue

if arr1[i]<arr2[j]:

i+=1

elif arr2[j]<arr1[i]:

j+=1

else:

# print(arr2[j],end=" ")

l.append(arr2[j])

flag=True

i+=1

j+=1

if flag is False:

l.append(arr2[j])

return l

import numpy as np

u=input().split(" ")

u1=np.array(u,dtype=np.str)

n=int(input())

r=np.char.multiply(u1,n).tolist()

p=' '.join(map(str,r))

u3=' '.join(map(str,u))

print(u3)

print(p)

**def** convert24(str1):

    # Checking if last two elements of time

    # is AM and first two elements are 12

**if** str1[**-**2:] **==** "AM" **and** str1[:2] **==** "12":

**return** "00" **+** str1[2:**-**2]

    # remove the AM

**elif** str1[**-**2:] **==** "AM":

**return** str1[:**-**2]

    # Checking if last two elements of time

    # is PM and first two elements are 12

**elif** str1[**-**2:] **==** "PM" **and** str1[:2] **==** "12":

**return** str1[:**-**2]

**else**:

        # add 12 to hours and remove PM

**return** str(int(str1[:2]) **+** 12) **+** str1[2:8]

# Driver Code

**print**(convert24("08:05:45 PM"))

x1=[1,2,3,4]In [17]:

np.repeat(x1,3)Out[17]:

array([1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4])

number\_str = str(a\_number)

Convert `a\_number` to a string

zero\_filled\_number = number\_str.zfill(5)

test\_string **=** 'GFG'

# printing original string

**print**("The original string : " **+** str(test\_string))

# No. of zeros required

N **=** 4

# using rjust()

# adding leading zero

res **=** test\_string.rjust(N **+** len(test\_string), '0')

# print result

print("The string after adding leading zeros : " **+** str(res))

m=int(input())

n=int(input())

m1=int(input())

n1=int(input())

h=[]

h1=[]

s=[]

for i in range(m):

l=list(map(int,input().split(" ")))

h.append(l)

for i in range(m1):

l=list(map(int,input().split(" ")))

h1.append(l)

if n!=m1:

print("Matrix multiplication not possible")

else:

for i in range(len(h)):

for j in range(len(h1[0])):

res=0

for k in range(len(h1)):

res=res+h[i][k]\*h1[k][j]

print(res,end=" ")

print(end="\n")

for perm in itertools.permutations([1,2,3,4,5,6], 4):

print(int("".join([str(i) for i in perm])))

#Biggest N digit integer that is divisible by P.

Integers N and P are given. Print the highest N-digit number divisible by P.

Example:

Input : N=3 , P=30

Output : 990

990 is the largest three digit

number divisible by 30.

Input : N = 2, P = 7

Output : 98

Input=

4

10

output=

9990

------------

Input=

3

9

output=

999

------------

Input=

4

2

output=

9998

Ans:-

n=int(input())

m=int(input())

i=10

max=i\*\*n-1

print(int(max-(max%m)))

import re

n=input()

h=re.compile('[@\_!#$%^&\*()<>?/\|}{~:]')

if h.search(n)!=None:

print("Yes")

else:

print("No")

from collections import Counter as c

m=input()

n=input()

d1=c(m)

d2=c(n)

comm=d1&d2

com=list(comm)

comm1=sorted(com)

print(len(comm))

print(' '.join(comm))

'''

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

'''

p=list(map(str,input().split(" ")))

q=input()

r=[]

for i in p:

for j in i:

l=" ".join(str(ord(j)))

r.append(l)

for i in range(len(r)):

if r[i]==q:

print(p[i])

'''

Welcome to GDB Online.

GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,

C#, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.

Code, Compile, Run and Debug online from anywhere in world.

'''

#print ('Hello World')

p=list(map(str,input().split(" ")))

q=input()

r=[]

for i in p:

g="".join(list(map(str,[ord(i[t])-96 for t in range(len(i))])))

r.append(g)

print(r)

for i in range(len(r)):

if r[i]==q:

print(p[i])

There is a lotterry with n coupons and n people take part of it. Each person picks exactly one coupon. Coupons are numbered consecutively

from 1 to n, and the coupon[i] has the number i written on it. The winner of the lottery is any person who owns a coupon with the sum of digits

written on it equal to s. If there are multiple winners, the prize is split equally among them. Determine how many values of s there are

where there is at least one winner and the prize is split among the most people.

For example, given the input n = 12, the list of ticket numbers is coupons = [1,2,3,4,5,6,7,8,9,10,11,12]. The sum of the digits are coupon\_sums =

[1,2,3,4,5,6,7,8,9,1,2,3]. The largest number of winners is 2 which will occur for tickets numbered [1,10],[2,11] and [3,12]. The maximum

number of possible winners occurs for any of 3 possible values of s, so 3 is the answer.

The program should print the number of ways to choose s, in such a way that there is at least one winner and the price is split among the

highest number of people.

The input contains an integer n denoting the maximum ticket number.

input = 3

output = 3

Explanation - THe three lottery coupons are numbered 1,2 and 3. The sum of the digits of the coupon numbers are 1,2 and 3 respectively.

There are three ways to choose s:

When s=1, only the person with coupon number = 1 is the winner.

When s=2, only the person with coupon number = 2 is the winner.

When s=3, only the person with coupon number = 3 is the winner.

input = 11

output = 2

Explanation - The lottery coupons are numbered from 1 to 11 and the sum of the digits of each of them is

1,2,3,4,5,6,7,8,9,1 and 2 respectively.

There are two ways to choose s:

When s =1, there are two winners and their coupon numbers are 1 and 10.

When s=2, there are two winners and their coupon numbers are 2 and 11.

input = 12

output = 3

input = 3

output = 3

input = 11

output = 2

n=int(input())

l=[sum(list(map(int,str(i)))) for i in range(1,n+1)]

l1={}

for i in l:

if i in l1:

l1[i]+=1

else:

l1[i]=1

r=all(x==1 for x in l1.values())

if r:

print(n)

else:

p=[i for i in l1.values() if i==max(list(l1.values()))]

print(len(p))

m=list(map(str,input().strip()))

m1=int(input())

m2=int(input())

l=[]

for i in range(m1):

a=m[0]

for j in range(0,len(m)-1):

m[j]=m[j+1]

m[len(m)-1]=a

for i in range(m2):

a=m[len(m)-1]

for j in range(len(m)-1,0,-1):

m[j]=m[j-1]

m[0]=a

print("".join(m))

You are given a set of releases of a software and you are asked to find the most recent release.

There may be multiple checkins of the software in a given branch. Each branch may also have

sub branches. For example release 3-5-4 refers to the fourth checkin in the fifth sub branch

of the third main branch. This hierarchy can go upto any number of levels.

If a level is missing it is considered as level 0 in that hierarchy. For example 3-5-7 is

same as 3-5-7-0 or even same as 3-5-7-0-0. The higher numbers denote more recent releases.

For example 3-5-7-1 is more recent than 3-5-7 but less recent than 3-6.

Input Format:

-------------

A single line space separated strings, list of releases

Output Format:

--------------

Print the latest release of the software.

Sample Input-1:

---------------

1-2 1-2-3-0-0 1-2-3

Sample Output-1:

----------------

1-2-3

Sample Input-2:

---------------

3-5-4 3-5-7 3-5-7-1 3-5-7-0-0 3-6

Sample Output-2:

----------------

3-6

import java.util.\*;

class Strings{

public static void main (String[] args) {

Scanner sc = new Scanner(System.in);

String arr[] = sc.nextLine().split(" ");

int maxlength=0;

Arrays.sort(arr,(o1,o2)->{

return o2.compareTo(o1);

});

String ans = arr[0];

out :for(int i=1;i<arr.length;i++){

String temp = "";

int k=0;

while(k<arr[i].length()){

if(arr[i].charAt(k) == ans.charAt(k)){

k++;

}

else break out;

}

int l=k;

while(k<ans.length()){

temp+="-0";

k+=2;

}

if(ans.substring(l).equals(temp)){ans=arr[i];}

else break;

}

System.out.println(ans);

}

}

/\*

You are asked to design the Tessellator scoring list.

The class is TessellatorScore.

The member functions are updatescore(studentid,score)

(if the student does not exist the student should be created in the scoring list

with score as his initial score).

Once the students gets an offer , he/she should be deleted using delete(studentid).

Before calling a company Sudheer Reddy – our placement officer wants to know

the sum of scores of top k students. So we must provide him a method getsum(k)

which returns the sum of of the scores of top k students.

Function calls:

===============

1. updatecore(int,int)

2. getsum(int)

3. delete(int)

0. stop the execution

Input Format:

-------------

There will be some lines of input,

Each line will startwith function call option and followed by parameters list

Output Format:

--------------

Print an integer when top funtion called.

Sample Input-1:

---------------

1 1 73

1 2 56

1 3 39

1 4 51

1 5 4

2 1

3 1

3 2

1 2 51

2 3

0

Sample Output-1:

----------------

73

141

Explanation:

------------

First line indicates 1 1 73, option 1 indicates updatescore(1,73) added to Score List.

After 5 lines of input, Score List will looks like

1 73

2 56

3 39

4 51

5 4

Next Line 2 1: Option 2 indicates getsum(1) answer is 73.

Next 2 lines 3 1, 3 2 indicates delete 1,2 from the Score List, new Score List is

3 39

4 51

5 4

Next line 1 2 51, option 1 -> update score and student id is 2 and score is 51

new Score List is:

2 51

3 39

4 51

5 4

Next line 2 3, Option 2 indicates getsum(3) answer is 141.

Next line 0, indicates stop the execution.

\*/

import java.util.\*;

class TessellatorScore {

Map<Integer, Integer> map;

TreeMap<Integer,Integer> sorted;

public TessellatorScore() {

map = new HashMap<>();

sorted = new TreeMap<>(Collections.reverseOrder());

}

public void updatescore(int playerId, int score) {

map.put(playerId,score);

sorted.put(score,sorted.getOrDefault(score,0)+1);

}

public int getsum(int K) {

int sum=0;

for(Map.Entry<Integer, Integer> key:sorted.entrySet()){

if(K<=0)break;

if(K >= key.getValue()){

sum += key.getKey()\*key.getValue();

K-=key.getValue();

}

else{

sum += key.getKey()\*K;

K=0;

}

}

return sum;

}

public void delete(int playerId) {

int marks = map.get(playerId);

map.remove(playerId);

sorted.put(marks,sorted.get(marks)-1);

}

}

public class Solution

{

public static void main(String args[])

{ Scanner sc = new Scanner(System.in);

// Your TessellatorScore object will be instantiated and called as such:

TessellatorScore ts = new TessellatorScore();

while(true){

int q = sc.nextInt();

if(q==0)return;

if(q==1){

ts.updatescore(sc.nextInt(),sc.nextInt());

}

else if(q==2){

System.out.println(ts.getsum(sc.nextInt()));

}

else{

ts.delete(sc.nextInt());

}

}

}

}

import java.util.\*;

class Strings{

static Set<String> set;

public static void main (String[] args) {

Scanner sc = new Scanner(System.in);

String words[] = sc.nextLine().split(" ");

set = new HashSet<>(Arrays.asList(words));

String num = sc.nextLine();

System.out.println(potential(num,""));

}

static String potential(String num,String formed){

//System.out.println(formed);

if(num.length() == 0){

if(set.contains(formed))return formed;

return "";

}

if(num.length()==1){

return potential(num.substring(1),formed+(char)(Integer.parseInt(num.substring(0,1))+'a'-1));

}

String ans1 = potential(num.substring(2),formed+(char)(Integer.parseInt(num.substring(0,2))+'a'-1));

String ans2 = potential(num.substring(1),formed+(char)(Integer.parseInt(num.substring(0,1))+'a'-1));

if(!ans1.equals(""))return ans1;

if(!ans2.equals(""))return ans2;

return "";

}

}

In this challenge you will create the username portion of a registation system.

All usernames must be unique. If a new user requests a name that is already used,

an integer should be added to the end of the username to make it unique.

The numbering begins with 1 and is incremented by 1 for each new instance per username.

As an example, if username requests were for [bob,bob,alice,alice,alice], your

system should assign usernames[bob,bo1,alice,alice1,alice2].

Given a list of username requests in the order given, process all requests and

return an array of the usernames as assigned by your function.

input =

4

alex

xylos

alex

alan

output =

alex

xylos

alex1

alan

Explanation

There are 4 usernames requested. Each of the usernames is unique with the exception of “alex”.

u2 =”alex” and must be modified. Since this is the first duplicate request for the username “alex”, your function should insert “alex1” to that position in u.

input =

4

alex

xylos

alex

alan

output =

alex

xylos

alex1

alan

input = 5

bob

bob

bob

alex

tom

output =

bob

bob1

bob2

alex

tom

input =

3

bob

tom

alex

output =

bob

tom

alex

input =

5

bob

tom

tom

bob

tom

output =

bob

tom

tom1

bob1

tom2

n=int(input())

l=[input() for i in range(n)]

p={}

for i in l:

if i in p:

p[i]+=1

else:

p[i]=1

p1=[]

for i in reversed(l):

n=p[i]

g=n-1

if g==0:

p1.append(i)

else:

p1.append(i+str(g))

p[i]=n-1

for i in reversed(p1):

print(i)

#Longest subsequence

"""

Write a python code using tuples to find length of

longest subsequence of the character 1. Without using Regular Expression.

Example:

Input=

0983111109384571111111111948584110

output=

10

case=1

input=245611158976345211111111111165879

output=12

-----------------------

case=2

input=47891113654811254

output=3

n=list(map(int,input().strip()))

p=[]

c=0

for i in range(len(n)-1):

if n[i]==1 and n[i]==n[i+1]:

c=c+1

else:

p.append(c)

c=0

continue

print(max(p)+1)

import math as m

n=list(map(int,input().split(" ")))

c=50

h=30

q=",".join([str(int(m.sqrt(m.ceil((2\*c\*i)/h)))) if i==1 else str(int(m.sqrt(((2\*c\*i)/h)))) for i in n ])

print(q)

If a filesystem has a block size of 4096 bytes, this means that a file comprised of only one byte will still use 4096 bytes of storage. A file made up of 4097 bytes will use 4096\*2=8192 bytes of storage. Knowing this, can you fill in the gaps in the calculate\_storage function below, which calculates the total number of bytes needed to store a file of a given size?

def calculate\_storage(filesize):

    block\_size = 4096

    # Use floor division to calculate how many blocks are fully occupied

    full\_blocks =4097

    # Use the modulo operator to check whether there's any remainder

    partial\_block\_remainder =filesize%block\_size

    # Depending on whether there's a remainder or not, return

    # the total number of bytes required to allocate enough blocks

    # to store your data.

    if partial\_block\_remainder > 0:

        return 4096\*(full\_blocks+1)

    return full\_blocks\*4096

print(calculate\_storage(1))    # Should be 4096

print(calculate\_storage(4096)) # Should be 4096

print(calculate\_storage(4097)) # Should be 8192

print(calculate\_storage(6000)) # Should be 8192

****Weight of fruits****

[Mathematical](https://practice.geeksforgeeks.org/topics/Mathematical/)

[Oracle](https://practice.geeksforgeeks.org/company/Oracle/)

If a fruit seller has initially 2000kg of watermelons in the morning out of which 99% is water. In the evening the watermelons shrink and the total water content is 98%. If none of the fruits are sold find the total weight of fruits.

Ans:-

99% of 2000 kg = 1980 kg water

and 20 kg of raw fruit.

In the evening, 2 % of the fruit was raw fruit (as 98 % was water). Let the fruit weight be x kgs.

(2/100) \*x =20

Therefore, x=1000 kgs

Ans:

x=int(input())

y=int(input())

z=int(input())

a=int(input())

y1=(x\*y)/100

z1=(x\*z)/100

x1=x-y1

q=(x1\*100\*a)//(100-z)

print(int(q))

#Cab and walk

Arun is working in an office which is N blocks away from his house.

He wants to minimize the time it takes him to go from his house to the office.

He can either take the office cab or he can walk to the office.

Arun's velocity is V1 m/s when he is walking. The cab moves with velocity V2 m/s but

whenever he calls for the cab, it always starts from the office, covers N blocks,

collects Arun and goes back to the office.

The cab crosses a total distance of N meters when going from office to Arun's house

and vice versa, whereas Arun covers a distance of (2–√∗N) while walking.

Help Arun to find whether he should walk or take a cab to minimize the time.

Input Format:

A single line containing three integer numbers N, V1, and V2 separated by a space.

Output Format:

Print 'Walk' or 'Cab' accordingly

Constraints:

1<=V1, V2 <=100

1<=N<=200

Example-1:

Input:

5 10 15

Output:

Cab

Example-2:

Input:

2 10 14

Output:

c

TestCases:

case=1

input=10 5 50

output=Cab

----------------

case=2

input=2 10 14

output=Walk

---------------

case=3

input=7 14 10

output=Walk

x,y,z=list(map(int,input().split(" ")))

t1=(x/y)

t2=(2\*\*0.5)\*x/z

if t1<=t2:

print("Walk")

else:

print("Cab")

m=int(input())

n=int(input())

c=0

for i in range(m+1):

for j in str(i):

if str(n) in j:

c=c+1

print(c)

Check if a given array is pairwise sorted or not

An array is considered pairwise sorted if each successive pair of numbers is in sorted (non-decreasing) order.

In case of odd elements, last element is ignored and result is based on remaining even number of elements.

The first input consists the size of the array followed by array elements

If pair wise sorted print "true" else "false"

Examples:

input = 6 10 15 9 9 1 5

output = true

Explanation

Pairs are (10, 15), (9, 9) and (1, 5).

All these pairs are sorted in non-decreasing order.

input = 6 10 15 8 9 10 5

output = false

Explanation

The last pair (10, 5) is not sorted.

input = 7 120 145 99 199 11 55 7

output = true

input = 1 10

output = true

input = 2 11 11

output = true

input = 7 120 145 99 199 11 55 7

output = true

input = 1 10

output = true

input = 2 11 11

output = true

input = 6 10 15 8 9 10 5

output = false

l=list(map(int,input().split(" ")))

c=0

for i in range(0,len(l)-1):

if l[i]<l[i+1]:

c=c+1

i=i+1

if c==int(len(l)//2):

print("true")

else:

print("false")

from collections import OrderedDict

m="".join(set(input()))

n="".join(set(input()))#if order dosent matter

"".join(OrderedDict.fromkeys(foo))#if order matters

c=0

for i in m:

if i in n:

c=c+1

if c>0:

print(c)

else:

print("-1")

for x in range(0,101,10):

Print(“{:>3}F|{:>6.2f} C”.format(x,(x-32)\*5/9))

### Formatting expressions

| **Expr** | **Meaning** | **Example** |
| --- | --- | --- |
| {:d} | integer value | '{:d}'.format(10.5) → '10' |
| {:.2f} | floating point with that many decimals | '{:.2f}'.format(0.5) → '0.50' |
| {:.2s} | string with that many characters | '{:.2s}'.format('Python') → 'Py' |
| {:<6s} | string aligned to the left that many spaces | '{:<6s}'.format('Py') → 'Py    ' |
| {:>6s} | string aligned to the right that many spaces | '{:>6s}'.format('Py') → '    Py' |
| {:^6s} | string centered in that many spaces | '{:^6s}'.format('Py') → '  Py ' |

**from** **string** **import** Template

s = Template('$who likes $what')

s.substitute(who='tim', what='kung pao')

'tim likes kung pao'

def convert\_seconds(n):

h=n//3600

m=(n-h\*3600)//60

s=n-h\*3600-m\*60

return h,m,s

print(convert\_seconds(3600))

for I,v in enumerate(win):

Print(“{} {}”.format(I+1,v))

list.clear()

List.copy()

List.extend(othr)

def format\_address(address\_string):

    num, st = address\_string.split(' ',1)

    return f"house number {num} on street named {st}"

def combine\_guests(guests1,guests2):

  combined\_dic = guests1

  for key2 in guests2:

    if key2 in guests1:

      pass

    else:

      combined\_dic[key2] = guests2[key2]

  return combined\_dic

To print common elements of two lists in java

S.o.p(l1.revealAll(l2)) // in order of elements of l1

Import java.util.\*;

Public static void main(String[]args){  
Scanner s=new Scanner(System.in);

Int x=s.nextInt();

Int arr[]=new arr[x];

S.o.pln(“ele of 0array”);

For(int I=0;I<x;I++){  
arr[I]=s.nextInt();

}

1. o.pln(“enter count of smallest repeating no”);

Int k=s.nextInt();

S.o.pln(count(arr,x,k));

}

Static int count(int arr[],int x,int k){

Arrays.sort(arr);

Int count=0;

For(int I=0;I<x;I++){

For(int j=I+1;j<x;j++){

If(arr[I]==arr[j]){

Count++;

}

If(count==k){

Return arr[I];

}

}

}

Return -1;

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Char a[]=a1.toCharArray();

Char b[]=a2.toCharArray();

Set<Character> sa=new HashSet<>();

Set<Character> sa=new HashSet<>();

For(int I=0;I<a.length;I++){

Sa.add(a[I]);

}

For(int I=0;I<b.length;I++){

Sb.add(b[I]);

}

Sa.retainAll(sb);

If(sa.size()>0){

T=”Yes”;

}

Else{

T=”NO”;

}

Retrun T;

}

(or)

For \_ in range(int(input()):

S1=set(input())

S2=set(input())

If s1.intersection(s2):

Print(“Yes”)

Else:

Print(“No”)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Int res=0;

Set<Integer> l=new HashSet<>();

For(int I:arr){

L.add(I);

}

For(int I:l){

If(l.contains(I+k){

Res++;

}

Return res;

}

Def f(s):

Return sum(I for c1,c2 in zip(s,s[1:]) if c1==c2)

T=int(input())

For \_ in range(t):

Print(f(input()))

#!/bin/python3

import os

import sys

import bisect

q = int(input())

n=[]

for q\_itr in range(q):

n.append(int(input()))

primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53]

it = 1

prods = []

for i in primes:

it \*= i

prods.append(it)

for i in n:

t = bisect.bisect(prods,i)

print(t)

#!/bin/python3

import math

import os

import random

import re

import sys

# Complete the pairs function below.

def pairs(k, arr):

    c=0

    seta=set(arr)

    for i in arr:

        if (i-k) in seta:

            c=c+1

    return c

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    N,K=(int(x) for x in sys.stdin.readline().split(" "))

    A=[int(x) for x in sys.stdin.readline().split(" ")]

    result = pairs(K,A)

    fptr.write(str(result) + '\n')

    fptr.close()