**Que 1:**

**Topic: Arrays**

**Problem Statement :** In a cricket match the coach wanted to check the performance of batsmen, So he decided the strike rate as criteria. He planned that the two batsmen whose strike rate difference is minimum will be sent no. 3 and no.4 in the next match. Now your task is to help the coach in finding the two batsmen.

**Input Format:**

You are given an integer ’n’ denoting the size of array. Next line contains n space separated integers.

**Output Format:**

Print the strike rate of two batsmen in same order of their occurence.

**Test Cases:**

Input 1:

6

138.3 156.5 156.6 160.2 198.3 146.2

Output 1:

156.5 156.6

Input 2:

5

122 365 245 122 456

Output 2:

122.0 122.0

Input 3:

4

1.25 2.35 3.14 1.22

Output 3:

1.25 1.22

Input 4:

5

22.35 35.25 35.12 25.0 10.66

Output 4:

35.25 35.12

Input 5:

6

44 22.5 44.00 12.45 36.11 7.2

Output 5:

44.0 44.0

Input 6:

5

78.2 98.2 77.1 33.2 78.15

Output 6:

78.2 78.15

Input 7:

5

4 2 4 5 3

Output 7:

4.0 4.0

Input 8:

7

10.2 11.2 10.9 45.2 14.2 33.2 66.2 65.2

Output 8:

10.2 10.9

Input 9:

4

10.2 34.5 56.7 10.34

Output 9:

10.2 10.34

Input 10:

5

84.12 15.22 36.22 15.27 11.2

Output 10:

15.22 15.27

**Solution:**

import math

n=int(input())

l=list(map(float,(input().split())))

lst=[]

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

    for j in range(i+1,len(l)):

        b=math.floor(l[i])

        a=math.floor(l[j])

        if(a==b):

            lst.append(l[i])

            lst.append(l[j])

print(\*lst)

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

**Que 2:**

**Topic: Arrays**

**Problem Statement:** Jack is fond of numbers. He has given an array.He has to find the maximum length of subarray such that sum of elements in that sub array gives output 0. He is in a bit of trouble, help him in finding a solution.

**Input Format:**

You are given an integer ‘n’ denoting the number of elements in array. Next line contains n space separated integers.

**Output Format:**

Print the max length of that array with sum 0.

**Test Cases:**

Input 1:

6

-4 3 1 0 0 6

Output 1:

5

Input 2:

5

1 0 0 0 1                                                                                                         Output 2:

3

Input 3:

4

1 -2  1 5

Output 3:

3

Input 4:

5

-20 12 8 0 1

Output 4:

4

Input 5:

6

11 12 -12 -11 14 12

Output 5:

4

Input 6:

5

1 0 0 3 5

Output 6:

2

Input 7:

5

4 2 -8 5 4

Output 7:

3

Input 8:

7

1 2 3 4 5 6 0

Output 8:

1

Input 9:

4

10 2 -12 0

Output 9:

4

Input 10:

5

8 -7 -1 10 9

Output 10:

3

**Solution:**

n=int(input())

a=list(map(int,input().split()))

sub=[]

for i in range(len(a)+1):

    for j in range(i+1,len(a)+1):

        s=a[i:j]

        if(sum(s)==0):

            sub.append(len(s))

print(max(sub))

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

**Que 3:**

**Topic: Strings**

**Problem Statement:** Given 2 strings S and A find the longest common substring.If no common substring is found print '-1'.

**Input Format:**

Take two strings as input.

**Output Format:**

Print the longest common substring, If no common substring is found print '-1'.

**Test Cases:**

Input 1:

code sold

Output 1:

od

Input 2:

compile compiler                                                                                                   Output 2:

compile

Input 3:

python language

Output 3:

n

Input 4:

value salute

Output 4:

alue

Input 5:

online offline

Output 5:

online

Input 6:

input output

Output 6:

put

Input 7:

Inside in

Output 7:

in out

-1

Input 8:

Format formatting

Output 8:

ormat

Input 9:

normal value

Output 9:

al

Input 10:

yes no

Output 10:

-1

**Solution:**

s1,s2=map(str,(input().split()))

l=""

for i in s1:

    if i in s2:

        l=l+i

if(len(l)==0):

    print(-1)

else:

    print(l)

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

**Que 4:**

**Topic: Strings**

**Problem Statement:** Given a string S of length N, find whether the given string is a palindrome or not and print 'yes' if string is a palindrome otherwise print 'no'.

**Input Format:**

Take string as an input.

**Output Format:**

Print 'yes' if string is a palindrome, otherwise print 'no'.

**Test Cases:**

Input 1:

compiler

Output 1:

no

Input 2:

madam                                                                                                   Output 2:

yes

Input 3:

debugger                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    Output 3:

no

Input 4:

ISAPALINILAPASI

Output 4:

yes

Input 5:

ABA

Output 5:

yes

Input 6:

input

Output 6:

not

Input 7:

ATOYOTA

Output 7:

yes

Input 8:

Format

Output 8:

no

Input 9:

javaj

Output 9:

yes

Input 10:

python

Output 10:

no

**Solution:**

s=input()

s1=s[::-1]

if(s==s1):

    print('yes')

else:

    print('no')

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

**Que 5:**

**Topic: Strings**

**Problem Statement:** Joseph was going through topic of strings. He learnt about anagrams. But due to some circumstances he forget ,now he hired you to help him in completing the work.Your task is to tell whether the two given strings are anagrams

**Input Format:**

The first line of the input is a string N, the second line of the input is a string M

**Output Format:**

Compare the two string input N and M. Print 1 if they are anagram else print 0.

**Test Cases:**

Input 1:

abcd

cdab

Output 1:

1

Input 2:

cat

act

Output 2:

1

Input 3:

triangle

integral                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    Output 3:

1

Input 4:

input

output

Output 4:

0

Input 5:

silent

listen

Output 5:

1

Input 6:

cases

spaces

Output 6:

0

Input 7:

file

life

Output 7:

1

Input 8:

programmer

compiler

Output 8:

0

Input 9:

java

python

Output 9:

0

Input 10:

abed

deba

Output 10:

1

**Solution:**

a=str(input())

b=str(input())

A=list(a)

A.sort()

B=list(b)

B.sort()

if(A==B):

    print(1)

else:

    print(0)

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

**Que 6:**

**Topic: Strings**

**Problem Statement:** Given 2 strings S1 and S2, work on the strings such that both string has the same number of characters.To adjust the length reduce number of exceeding characters from longer string.

**Input Format:**

Take two strings as an input.

**Output Format:**

Print the string with reduced number of exceeding characters from longer string.

**Test Cases:**

Input 1:

 guvi geeks

Output 1:

guvigeek

Input 2:

talentio trainer

Output 2:

talentitrainer

Input 3:

triangle format                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    Output 3:

triangformat

Input 4:

input output

Output 4:

inputoutpu

Input 5:

silent language

Output 5:

silentlangua

Input 6:

test cases

Output 6:

testcase

Input 7:

normal solution

Output 7:

normalsoluti

Input 8:

programmer compiler

Output 8:

programmcompiler

Input 9:

python java

Output 9:

pythjava

Input 10:

insert delete

Output 10:

insertdelete

**Solution:**

s1,s2=map(str,input().split())

l1=len(s1)

l2=len(s2)

if(l1==l2):

    print(s1,end="")

    print(s2)

if(l1>l2):

    l=l1-l2

    print(s1[:-l],end="")

    print(s2)

if(l2>l1):

    L=l2-l1

    print(s1,end="")

    print(s2[:-L])

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

**Que 7:**

**Topic: Strings**

**Problem Statement:** Given an arraylist A of string type which has name#mark1#mark2#mark3 format. Retrieve the name of the student who has scored max marks(total of three).

**Input Format:**

Take string as an input.

**Output Format:**

Print the name of the student who has scored maximum marks.

**Test Cases:**

Input 1:

arun#12#12#12

deepak#13#12#12

Output 1:

deepak

Input 2:

aman#14#16

amol#12#10

Output 2:

aman

Input 3:

nandini#19#12#11

mrunal#20#12#11                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  Output 3:

mrunal

Input 4:

aryan#25#12#78

ayan#78#12#24

Output 4:

aryan

Input 5:

sahil#100#98#97

saniya#58#64#78

Output 5:

sahil

Input 6:

sheeba#78#12#45

arif#72#14#45

Output 6:

sheeba

Input 7:

prajakta#100#87#79

shrawani#97#79#98

Output 7:

shrawani

Input 8:

programmer#98#147#23

sportsman#25#47#100

Output 8:

programmer

Input 9:

compiler#10#20#30

debugger#12#22#33

Output 9:

debugger

Input 10:

mahider#12#78#56

shubham#15#24#26

Output 10:

mahider

**Solution:**

import re

s1=input()

s2=input()

a1=re.findall(r'\d+',s1)

d1=list(map(int,a1))

n1=re.findall(r'\w+',s1)

name1=list(map(str,n1))

a2=re.findall(r'\d+',s2)

d2=list(map(int,a2))

n2=re.findall(r'\w+',s2)

name2=list(map(str,n2))

t1=sum(d1)

t2=sum(d2)

if(t1>t2):

    print(name1[0])

else:

    print(name2[0])

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

**Que 8:**

**Topic: Strings**

**Problem Statement:** You are given a string ‘s’, find the minimum number of characters to be inserted to convert it to a palindrome.

**Input Format:**

You are given a string ‘s’.

**Output Format:**

Print 0 if no extra character is required else print the amount of characters required.

**Test Cases:**

Input 1:

abcd

Output 1:

3

Input 2:

value

Output 2:

4

Input 3:

python                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        Output 3:

5

Input 4:

madam

Output 4:

0

Input 5:

abab

Output 5:

1

Input 6:

form

Output 6:

3

Input 7:

nandini                                                                                                                                                                                                                  Output 7:

4

Input 8:

banana

Output 8:

1

Input 9:

aba

Output 9:

0

Input 10:

racecar

Output 10:

0

**Solution:**

def m(a,b):

    return min(a,b)

def Mininsertion(s,n):

    table=[[0 for i in range(n)]

              for i in range(n)]

    l,h,g=0,0,0

    for g in range(1,n):

        l=0

        for h in range(g,n):

            if(s[l]==s[h]):

                table[l][h]=table[l+1][h-1]

            else:

                table[l][h]=(m(table[l][h-1],table[l+1][h])+1)

            l=l+1

    return table[0][n-1];

s=input()

n=len(s)

print(Mininsertion(s,n))

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

**Que 9:**

**Topic: Strings**

**Problem Statement:** Given a string S consisting of lowercase latin letters, arrange all its letters in lexicographical order using Counting Sort.

**Input Format:**

You are given a string ‘s’.

**Output Format:**

Print the sorted string.

**Test Cases:**

Input 1:

talentio

Output 1:

aeilnott

Input 2:

Programmer

Output 2:

aegmmoprrr

Input 3:

python

Output 3:

hnopty

Input 4:

format

Output 4:

afmort

Input 5:

debugger

Output 5:                                                                                                                                                                                    bdeeggru

Input 6:

form

Output 6:

fmor

Input 7:

static                                                                                                                                                                                                                  Output 7:

acistt

Input 8:

dynamic

Output 8:

acdimny

Input 9:

online

Output 9:

eilnno

Input 10:

offline

Output 10:

effilno

**.Solution:**

s=input()

l=sorted(s)

new=''

for i in l:

  new=new+i

print(new)

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

**Que 10:**

**Topic: Strings**

**Problem Statement:** Given a string S, remove the characters which exist more than one times,and print the remaining string.

**Input Format:**

You are given a string ‘s’.

**Output Format:**

Print the string without characters that are repeated.

**Test Cases:**

Input 1:

trigger

Output 1:

tie

Input 2:

Programmer

Output 2:                                                                                                                 pogae

Input 3:

python

Output 3:

python

Input 4:

Engineering

Output 4:

Er

Input 5:

debugging

Output 5:

debuin

Input 6:

solution

Output 6:

slutin

Input 7:

memory                                                                                                                                                                                                                 Output 7:

eory

Input 8:

management

Output 8:

gt

Input 9:

online

Output 9:

olie

Input 10:

offline

Output 10:

oline

**Solution:**

import re

s=input()

a=re.findall(r'\w',s)

d=list(map(str,a))

l=[]

for i in d:

    if(d.count(i)==1):

        l.append(i)

r=''.join(l)

print(r,end="")

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

**Que 11:**

**Topic: Strings**

**Problem Statement:** Given two strings S1 and S2,display 'yes' if given two strings are complementary otherwise display 'no'. If we join alphabets of both the strings we should get all 26 capital letters exactly once, then only we can call them as complementary.

**Input Format:**

You are given two strings ‘s1’ and ‘s2’.

**Output Format:**

Print 'yes' if given two strings are complementary, otherwise print 'no'.

**Test Cases:**

Input 1:

ABDCFGIJKLMNOPQUVWXYZ

EHRST

Output 1:

yes

Input 2:

ABCDFGHIJKLMNOPQRTUVWXYZ

ES

Output 2:

 yes

Input 3:

ABCDEFGHIJ

KLM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    Output 3:

no

Input 4:

ABCDEFGHIJKLMNOP

QRSTUVWXYZ

Output 4:

yes

Input 5:

ABCDEFGNOPQRSTUVWXYZ

HIJKLM

Output 5:

yes

Input 6:

ABCDE

FGH

Output 6:

no

Input 7:

AGHIJKLMNOPQRSTUVWXY

BCDEFZ                                                                                                                                                                                                                                                                                                 Output 7:

yes

Input 8:

ABCDEFGH

IJKLMNOPQRSTUVWXYZ

Output 8:

yes

Input 9:

ABCD

XYZ

Output 9:

no

Input 10:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDE

Output 10:

no

**Solution:**

s1=input()

s2=input()

s3=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

s4=s1+s2

S=list(s4)

S.sort()

if(S==s3):

    print('yes')

else:

    print('no')

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

**Que 12:**

**Topic: Strings**

**Problem Statement:** Roman jhon want to identify the repeated letters in two given strings and capitalize it.Help him to achieve it.

**Input Format:**

You are given two strings ‘s1’ and ‘s2’.

**Output Format:**

Print the string with capitalized repeated letters.

**Test Cases:**

Input 1:

programming language

Output 1:

pRoGRAMMiNG lANGuAGe

Input 2:

online gdb                                                                                                                                                                                                                                                                                                          Output 2:

oNliNe gdb

Input 3:

talentio trainer                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           Output 3:

TAlENTIo TRAINER

Input 4:

python questions

Output 4:

pyThON queSTiONS

Input 5:

count number                                                                                                                                                                                                                                                                                                                      Output 5:

coUNt NUmber

Input 6:

append command

Output 6:

APPeND coMMAND

Input 7:

my project                                                                                                                                                                                                                                                                                      Output 7:

my project

Input 8:

learn python

Output 8:

learN pythoN

Input 9:

string outline

Output 9:

sTrINg ouTlINe

Input 10:

good coder

Output 10:

gOOD cODer

**Solution:**

s1,s2=map(str,input().split())

s3=s1+" "+s2

s4=[]

for i in s3:

    if(s3.count(i)>1):

        a=i.capitalize()

        s4.append(a)

    if(s3.count(i)==1):

        s4.append(i)

l=''.join(s4)

print(l)

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

**Que 13:**

**Topic: Strings**

**Problem Statement:** Given a string and a number K, change every kth character to uppercase from beginning in string, also if the string contains character which is same to the kth character then change it too uppercase.

**Input Format:**

You are given a string ‘s’ and position ‘k’.

**Output Format:**

Print the string with every kth position character in uppercase.

**Test Cases:**

Input 1:

python 2

Output 1:

pYtHoN

Input 2:

online 3                                                                                                                                                                                                                                                                                                                                                                                                            Output 2:

onLinE

Input 3:

append 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   Output 3:

aPPEnD

Input 4:

debugger 4

Output 4:

debUggeR

Input 5:

input 1                                                                                                                                                                                                                                                                                                                     Output 5:

 INPUT

Input 6:

format 3

Output 6:

foRmaT

Input 7:

learning 2                                                                                                                                                                                                                                                                                                                                                                                            Output 7:

lEaRnInG

Input 8:

languages 5

Output 8:

langUages

Input 9:

tutorials 2

Output 9:

tUtOrIaLs

Input 10:

operation 3

Output 10:

opEraTioN

**Solution:**

s,n=input().split(maxsplit=1)

k=int(n)

l=list(s)

for i in(k,len(l)):

    f=l[(k-1)::k]

a=[]

for i in l:

    if i in f:

        a.append(i)

b=[]

for j in l:

    if j in a:

        d=str(j)

        d1=d.upper()

        b.append(d1)

    if j not in a:

        d2=str(j)

        b.append(d2)

e="".join(b)

print(e)

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

**Que 14:**

**Topic: Strings**

**Problem Statement:** Indian PAN card issuing authority have found some fake PAN cards. They have hired you so that you can validate PAN card for them. Your task is to develop a suitable algorithm which could check if pan is valid or not

1) Pan must have uppercase letters only.

2) It must be of 10 characters only

3) From index 1 to 5 all must be letters(A-Z),last index must be letter

4) Rest all must be integer Starting from 1

**Input Format:**

You are given an input string which indicates the PAN number

**Output Format:**

Print 'pan' if it is valid PAN number, else print 'not pan'

**Test Cases:**

Input 1:

HXTPS2142R

Output 1:

pan

Input 2:

CVDET1234J                                                                                                                                                                                                                                                                                                                                                                                                            Output 2:

pan

Input 3:

MRUNA4567L                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       Output 3:

pan

Input 4:

debu498gh

Output 4:

not pan

Input 5:

JODRT4563J                                                                                                                                                                                                                                                                                                                    Output 5:

 pan

Input 6:

JAYUK9974O

Output 6:

pan

Input 7:

learf456lk                                                                                                                                                                                                                                                                                                                                                                                            Output 7:

not pan

Input 8:

564MNRTY5L

Output 8:

not pan

Input 9:

ZYXSD9870L

Output 9:

pan

Input 10:

ASDEA2456R

Output 10:

pan

**Solution:**

n=input()

l=len(n)

if(l==10):

    if(n[0:5].isupper()):

        if(n[5:8].isdigit()):

            if(n[9].isupper()):

                print('pan')

            else:

                print('not pan')

        else:

            print('not pan')

    else:

        print('not pan')

else:

    print('not pan')

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

**PRIYA (TOPIC : ARRAYS)**

**==================================================**

**Ques:15**

Arvind Kejriwal wants to implement a revised version of the ODD/EVEN scheme for vehicles in order to reduce pollution in Delhi. The vehicle numbers are stored in an array and instead of allowing only either of odd/even numbered vehicles on the road, he decides to allow the vehicles at odd/even INDICES of the array on a particular day. The day being Monday, only vehicles at even indices are allowed on the road. He wants to analyse the number of vehicles that are allowed on the road on that particular day and hence decides to order the data in ascending order. Sort the elements at the even indices of the array in ascending order and help make the IITan’s idea a success! Kejriwal would be happy if you could do it in O(Nlog N) complexity, where N is the number of elements that are to be sorted.

Input Description:

Size of the array followed by the elements of the array

Output Description:

Array with elements at the even indexes sorted in ascending order.

Sample Input :

6

12 44 23 16 1 62

Sample Output :

1 44 12 16 23 62

**Test Cases:**

Input 1:

15

15 48 75 96 62 32 20 41 32 35 78 95 63 20 11

Output 1:

11 48 15 96 20 32 32 41 62 35 63 95 75 20 78

Input 2:

10

4 7 5 2 6 3 2 1 5 9

                                                                                                                                                                                                                                                                                                                                                                                                       Output 2:

 2 7 4 2 5 3 5 1 6 9

Input 3:

 9

45 75 86 52 14 6 21 84 95

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        Output 3:

 14 75 21 52 45 6 86 84 95

Input 4:

 6

14 25 41 63 21 30

Output 4:

 14 25 21 63 41 30

Input 5:

 8

15 42 36 78 45 96 21 30

                                                                                                                                                                                                                                                                                                         Output 5:

 15 42 21 78 36 96 45 30

Input 6:

8

1 2 3 9 5 8 7 6

Output 6:

1 2 3 9 5 8 7 6

Input 7:

7

14 75 95 63 21 45 20                                                                                                                                                                                                                                                                                                                                                                                           Output 7:

14 75 20 63 21 45 95

Input 8:

 5

10 15 24 632 152

Output 8:

 10 15 24 632 152

Input 9:

6

12 44 23 16 1 62

Output 9:

1 44 12 16 23 62

Input 10:

3

12 42 51

Output 10:

 12 42 51

**Solution:**

n=int(input())

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

a=[]

b=[]

for i in range(len(l)):

    if(i%2==0):

        a.append(l[i])

    else:

        b.append(l[i])

c=[]

a.sort()

j=0

for j in range(len(b)):

    c.append(a[j])

    c.append(b[j])

    j=j+1

if(len(a)>len(b)):

    c.append(a[-1])

print(\*c)

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

**Ques:16**

You are given with arrays containing some non negative integers.Your task is to make the largest number out of it.

Input Description:

First line contains the size of array ’n’. Second line contains the n space separated integers.

Output Description:

Print the largest element that can be form out of the array

Sample Input :

6

12 19 546 60

Sample Output :

605461912

**Test Cases:**

Input 1:

 7

45 12 6 32 14 75 85

Output 1:

 8575645321412

Input 2:

7

12 14 52 63 21 20 47                                                                                                                                                                                                                                                                                                                                                                                                           Output 2:

63524721201412

Input 3:

5

14 74 58 96 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 Output 3:

9674583214

Input 4:

 6

12 47 85 96 23 15

Output 4:

968547231512

Input 5:

 4

152 475 963 215                                                                                                                                                                                                                                                                                                                  Output 5:

 963475215152

Input 6:

 8

41 526 37 85 96 42 62 10

Output 6:

 96856252642413710

Input 7:

7

4 7 5 8 9 6 3                                                                                                                                                                                                                                                                                                                                                                                           Output 7:

9876543

Input 8:

4

12 51 526 34

Output 8:

526513412

Input 9:

5

10 12 15 43 62

Output 9:

6243151210

Input 10:

6

12 19 546 60

Output 10:

605461912

**Solution:**

from itertools import permutations

n=int(input())

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

lst = []

for i in permutations(l, len(l)):

    lst.append("".join(map(str,i)))

print(max(lst))

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

**Ques:17**

Given 2 numbers N,Q and an array of N integers, followed by Q queries each consisting of L,R. Find the GCD of numbers in [L,R]. If there is only one number in the given range, then print the number itself.

Input Size : 1<=N<=1000

Example:

INPUT

5 2

1 2 3 4 5

1 2

4 5

OUTPUT

1

1

**Test Cases:**

Input 1:

 7 2

1 4 7 5 9 6 2

11 25

12 2

Output 1:

 1

2

Input 2:

4 4

1 2 3 4

12 3 44 52

56 23 14 52

47 45 96 23

14 12 153 32                                                                                                                                     Output2:

 3

1

1

2

Input 3:

 3 2

14 25 36

12 15

21 20                                                                                                                                                     Output 3:

 3

1

Input 4:

4 4

1 4 7 5

2 5 6 3

1 5 10 15

10 12 14 16

15 32 45 75

Output 4:

1

1

2

1

Input 5:

4 5

4 7 5 2

1 5 3 7 8

2 4 6 8 10

3 15 21 24 30

4 1 5 2 6

7 15 14 23 20

                                                                                                                                                               Output 5:

1

2

3

1

1

Input 6:

6 3

4 7 5 9 6 2

1 4 6

4 8 12

3 12 15

Output 6:

1

4

3

Input 7:

5 2

1 4 7 5 6

2 6

3 9                                                                           Output 7:

2

3

Input 8:

3 3

2 4 6

1 3 5

4 8 10

2 4 8

Output 8:

1

4

2

Input 9:

3 4

4 5 2

4 7 5 9

4 5 6 9

1 2 3 4

4 1 2 0

Output 9:

1

1

1

1

Input 10:

2 2

10 12

20 30

15 20

Output 10:

10

5

**Solution:**

import math

def gc(x,y):

    while(y):

        x,y=y,x%y

    print(x)

n,q=map(int,input().split())

lst=list(map(int,input().split()))

for i in range(q):

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

    a=len(l)

    if(a==1):t(\*l)

    else:

        x,y=l[0],l[1]

        gc(x,y)

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

**Ques:18(575)**

**Assume you are a student studying in school.You are given a task to find first negative integer for each and every window of size k.**

**Input Description:**

**First line contains an integer n denoting the size of the array. The next line contains n space separated integers forming the array. The last line contains the window size k.**

**Output Description:**

**Print the first negative integer in that window.If all the numbers are positive print 0**

**Sample Input :**

**7**

**1 -2 -3 -4 5 6 -7**

**3**

**Sample Output :**

**-2 -3 -4 -7**

**Test Cases:**

**Input 1:**

**5**

**-7 -8 5 6 -2**

**1**

**Output 1:**

**-7 -8 -2**

**Input 2:**

**7**

**-7 5 -9 -2 -3 -1 2**

**4**

**Output 2:**

**-7 -9 -9 -2**

**Input 3:**

**4**

**-1 -5 -7 -2**

**3**

**Output 3:**

**-1 -5**

**Input 4:**

**8**

**4 5 -2 -6 -8 -4 -5 2**

**5**

**Output 4:**

**-2 -2 -2 -6**

**Input 5:**

**8**

**4 -2 -1 -6 -12 -8 4 6**

**2**

**Output 5:**

**-2 -2 -1 -6 -12 -8 0**

**Input 6:**

**7**

**1 4 -7 -5 -6 2 3**

**5**

**Output 6:**

**-7 -7 -7**

**Input 7:**

**6**

**-2 -1 -7 5 6 2**

**2**

**Output 7:**

**-2 -1 -7 0**

**Input 8:**

**5**

**7 4 -8 -9 -5**

**3**

**Output 8:**

**-8 -8 -8**

**Input 9:**

**6**

**-4 -8 5 2 6 3**

**2**

**Output 9:**

**-4 -8 0**

**Input 10:**

**7**

**-4 -5 -6 -8 9 2 3**

**4**

**Output 10:**

**-4 -5 -6 -8**

**Solution:**

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

**a=list(map(int,(input().split())))**

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

**l=[]**

**for i in range(0, (n - k + 1)):**

**flag = False**

**for j in range(0, k):**

**if (a[i + j] < 0):**

**l.append(a[i+j])**

**flag = True**

**break**

**if (not(flag)):**

**l.append('0')**

**print(\*l)**

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

**Ques:19(554)**

**You are given with two arrays. Your task is to merge the array such that first array is in ascending order and second one in descending order.**

**Input Description:**

**First line contains two integer ‘n’ and ‘m’. ‘n’ denotes length of array 1 and ‘m’ of array 2.Next line contains n space separated numbers and third line contains ‘m’ space separated numbers**

**Output Description:**

**Print a single array in desired order**

**Sample Input :**

**3 3**

**23 15 16**

**357 65 10**

**Sample Output :**

**15 16 23 357 65 10**

Test Cases:

Input 1:

5 5

41 52 63 24 75

54 85 96 523 14

Output 1:

24 41 52 63 75 523 96 85 54 14

Input 2:

4 7

4 5 2 6

4 8 5 6 3 2 1

Output 2:

2 4 5 6 8 6 5 4 3 2 1

Input 3:

4 4

1 5 4 2

5 6 3 2

Output 3:

1 2 4 5 6 5 3 2

Input 4:

4 5

4 7 8 5

1 5 6 3 24

Output 4:

4 5 7 8 24 6 5 3 1

Input 5:

4 5

14 52 32 15

47 85 96 53 21

Output 5:

14 15 32 52 96 85 53 47 21

Input 6:

3 2

15 20 34

10 20

Output 6:

15 20 34 20 10

Input 7:

5 6

1 4 5 7 2

3 2 5 4 6 8

Output 7:

1 2 4 5 7 8 6 5 4 3 2

Input 8:

4 5

1 4 7 52

56 23 14 58 63

Output 8:

1 4 7 52 63 58 56 23 14

Input 9:

3 6

47 45 21

24 63 15 75 89 63

Output 9:

21 45 47 89 75 63 63 24 15

Input 10:

5 3

41 75 96 52 34

15 23 24

Output 10:

34 41 52 75 96 24 23 15

**Solution:**

**n,m=map(int,(input().split()))**

**l1=list(map(int,(input().split())))**

**l2=list(map(int,(input().split())))**

**l1.sort()**

**l2.sort(reverse=True)**

**print(\*l1,end=" ")**

**print(\*l2)**

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

**Ques:20(644)**

**You are given a number with duplicate digits your task is to remove the immediate duplicate digits and print the result**

**Input Description:**

**You are given a long string of digits**

**Output Description:**

**Print the desired result print -1 if result length is 0**

**Sample Input :**

**1331**

**Sample Output :**

**-1**

**Test Cases:**

**Input 1:**

**14425**

**Output 1:**

**125**

**Input 2:**

**47562**

**Output 2:**

**47562**

**Input 3:**

**44751126**

**Output 3:**

**7526**

**Input 4:**

**1121163**

**Output 4 :**

**263**

**Input 5:**

**478593212**

**Output 5:**

**4785931**

**Input 6:**

**114452663**

**Output 6:**

**523**

**Input 7:**

**12211235**

**Output 7:**

**35**

**Input 8:**

**114752633**

**Output 8:**

**47526**

**Input 9:**

**1152663247**

**Output 9:**

5347

**Input 10:**

**445522633**

**Output 10:**

**6**

**Solution:**

**import re**

**s=input()**

**b=""**

**temp = re.findall(r'\d',s)**

**d= list(map(str, temp))**

**new=""**

**for i in d:**

**if(d.count(i)==1):**

**new=new+i**

**if(len(new)==0):**

**print('-1')**

**else:**

**print(new)**

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

**Ques:21(602)**

**You are given with an array of numbers, Your task is to print the difference of indices of largest and smallest number.All number are unique.**

**Input Description:**

**First line contains a number ‘n’. Then next line contains n space separated numbers.**

**Output Description:**

**Print the difference of indices of largest and smallest array**

**Sample Input :**

**5**

**1 6 4 0 3**

**Sample Output :**

**-2**

**Test Cases:**

**Input 1:**

**5**

**47 45 32 663 1**

**Output 1:**

**-1**

**Input 2:**

**6**

**4 7 5 2 6 3**

**Output 2:**

**-2**

**Input 3:**

**7**

**47 52 63 21 452 74 56**

**Output 3:**

**1**

**Input 4:**

**8**

**12 45 261 358 75 63**

**Output 4:**

**3**

**Input 5:**

**6**

**4 7 8 5 9 4**

**Output 5:**

**4**

**Input 6:**

**4**

**7 4 5 2**

**Output 6:**

**-3**

**Input 7:**

**6**

**7 4 5 8 23 1**

**Output 7:**

**-1**

**Input 8:**

**5**

**2 4 5 1 3**

**Output 8:**

**-1**

**Input 9:**

**6**

**1 4 2 6 3 2**

**Output 9:**

**3**

**Input 10:**

**9**

**1 4 7 5 9 6 40 0 23**

**Output 10:**

**-1**

**Solution:**

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

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

**d=min(l)**

**s=max(l)**

**la=l.index(s)**

**sm=l.index(d)**

**a=la-sm**

**print(a)**

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

**Ques:22(687)**

**You are given an array of numbers. Print the least occurring element. If there is more than 1 element print all of them in decreasing order of their value.**

**Input Description:**

**You are given a number ‘n’ denoting size of array. Next line contains n space separated numbers.**

**Output Description:**

**Print the number as mentioned**

**Sample Input :**

**9**

**1 6 4 56 56 56 6 4 2**

**Sample Output :**

**2 1**

**Test Cases:**

**Input 1:**

**6**

**12 41 51 41 52 12**

**Output 1:**

**52 51**

**Input 2:**

**6**

**12 41 52 63 21 42**

**Output 2:**

**63 52 42 41 21 12**

**Input 3:**

**5**

**12 41 53 62 68**

**Output 3:**

**68 62 53 41 12**

**Input 4:**

**8**

**42 6321 542 632 57 859 631 425**

**Output 4:**

**6321 859 632 631 542 425 57 42**

**Input 5:**

**4**

**75 485 632 415**

**Output 5:**

**632 485 415 75**

**Input 6:**

**7**

**151 142 5632 154 251 251 151**

**Output 6:**

**5632 154 142**

**Input 7:**

**7**

**45 75 63 21 52 63 54**

**Output 7:**

**75 54 52 45 21**

**Input 8:**

**6**

**11 425 63 214 52 63**

**Output 8:**

**425 214 52 11**

**Input 9:**

**9**

**4 5 2 6 3 1 1 7 2**

**Output 9:**

**7 6 5 4 3**

**Input 10:**

**5**

**4 1 4 1 2**

**Output 10:**

**2**

**Solution:**

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

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

**s=[]**

**for i in l:**

**if(l.count(i)==1):**

**s.append(i)**

**s.sort(reverse=True)**

**print(\*s)**

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

**Ques:23(553)**

**You are passport issuer, but due to some problem in the system there are redundant  passport numbers. Your task is to delete all the duplicate passport numbers.You are given with a list of passport numbers.**

**Input Description:**

**You are given length of list.Second line,You are given with a list**

**Output Description:**

**Print the list of passport numbers without duplicates**

**Sample Input :**

**5**

**A23 B56  B56 C79 D16**

**Sample Output :**

**A23 B56 C79 D16**

**Test Cases:**

**Input 1:**

**6**

**A15 B16 C24 D75 R85 B86**

**Output 1:**

**A15 B16 C24 D75 R85 B86**

**Input 2:**

**5**

**A15 A16 A23 B15 B14**

**Output 2:**

**A15 A16 A23 B15 B14**

**Input 3:**

**4**

**A15 C15 D26 E23**

**Output 3:**

**A15 C15 D26 E23**

**Input 4:**

**5**

**A15 V15 C16 C12 23**

**Output 4:**

**A15 V15 C16 C12 23**

**Input 5:**

**6**

**A15 A15 A15 B16 C23 D41**

**Output 5:**

**A15 B16 C23 D41**

**Input 6:**

**8**

**A15 A26 B15 V62 A15 A26 B15 V62**

**Output 6:**

**A15 A26 B15 V62**

**Input 7:**

**7**

**A74 A15 A74 B45 C63 D24 E15**

**Output 7:**

**A74 A15 B45 C63 D24 E15**

**Input 8:**

**6**

**A14 A14 A15 B16 C23 C45**

**Output 8:**

**A14 A15 B16 C23 C45**

**Input 9:**

**5**

**A21 B47 B17 C45 B47**

**Output 9:**

**A21 B47 B17 C45**

**Input 10:**

**5**

**A26 A35 A48 B51 C63**

**Output 10:**

**A26 A35 A48 B51 C63**

**Solution:**

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

**s=list(map(str,(input().split())))**

**l=[]**

**for i in s:**

**if i not in l:**

**l.append(i)**

**print(\*l)**

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

**Ques:24(688)**

**Pk finds it difficult to judge the minimum element in the list of elements given to him. Your task is to develop the algorithm in order to find the minimum element.**

**Note:Don’t use sorting**

**Input Description:**

**You are given ‘n’ number of elements. Next line contains n space separated numbers.**

**Output Description:**

**Print the minimum element**

**Sample Input :**

**5**

**3 4 9 1 6**

**Sample Output :**

**1**

**Test Cases:**

**Input 1:**

**8**

**14 526 321 14 526 32 20 14**

**Output 1:**

**14**

**Input 2:**

**6**

**47 58 62 32 12 10**

**Output 2:**

**10**

**Input 3:**

**7**

**4 15 2 6 32 41 2**

**Output 3:**

**2**

**Input 4:**

**5**

**12 4 57 596 32**

**Output 4:**

**4**

**Input 5:**

**6**

**47 11 52 63 24 52**

**Output 5:**

**11**

**Input 6:**

**7**

**45 12 63 24 6 87 3**

**Output 6:**

**3**

**Input 7:**

**3**

**15 20 42**

**Output 7:**

**15**

**Input 8:**

**4**

**4 1 5 2**

**Output 8:**

**1**

**Input 9:**

**5**

**4 2 6 3 2**

**Output 9:**

**2**

**Input 10:**

**6**

**4 1 5 2 6 3**

**Output 10:**

**1**

**Solution:**

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

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

**print(min(l))**

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

**Ques:25(597)**

**you are given with array of numbers.you have to find whether array is beautiful or not. A beautiful array is an array whose sum of all numbers is divisible by 2, 3 and 5**

**Input Description:**

**You are given a number ‘n’ denoting the size of array.Next line contains n space separated numbers.**

**Output Description:**

**Print 1 if array is beautiful and 0 if it is not**

**Sample Input :**

**5**

**5 25 35 -5 30**

**Sample Output :**

**1**

**Test Cases:**

**Input 1:**

**8**

**4 1 2 3 6 4 7 2**

**Output 1:**

**0**

**Input 2:**

**4**

**4 5 6 2**

**Output 2:**

**0**

**Input 3:**

**5**

**12 24 25 26 30**

**Output 3:**

**0**

**Input 4:**

**3**

**15 24 32**

**Output 4:**

**0**

**Input 5:**

**5**

**25 35 -25**

**Output 5:**

**0**

**Input 6:**

**5**

**25 35 -25 30 5**

**Output 6:**

**0**

**Input 7:**

**5**

**5 25 -4 30**

**Output 7:**

**0**

**Input 8:**

**3**

**10 20 30**

**Output 8:**

**1**

**Input 9:**

**4**

**15 20 36 4**

**Output 9:**

**0**

**Input 10:**

**6**

**12 3 15 20 30 10**

**Output 10:**

**1**

**Solution:**

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

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

**s=sum(l)**

**if(s%2==0 and s%3==0 and s%5==0):**

**print(1)**

**else:**

**print(0)**

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

**Ques:26(493)**

**Ramesh is a student and wants to find out if there is any other student in his class who has got the same marks as his, in maths. Help him to find out.**

**Input Description:**

**First line contains the number of students in the class followed by Ramesh’s mark. Second line contains the marks of all students in the class.**

**Output Description:**

**Index of student who got mark same as Ramesh’s mark. If no such mark exists, return -1.**

**Sample Input :**

**2 10**

**1 2**

**Sample Output :**

**-1**

**Test Cases:**

**Input 1:**

**4 2**

**1 5 4 2 6 3 2**

**Output 1:**

**3**

**Input 2:**

**6 5**

**4 2 1 5 6 3 2**

**Output 2:**

**3**

**Input 3:**

**4 2**

**1 4 5 7 2 8 6 32**

**Output 3:**

**4**

**Input 4:**

**5 5**

**1 4 2 6 32 5**

**Output 4:**

**5**

**Input 5:**

**4 1**

**4 5 1 2 6 5475**

**Output 5:**

**2**

**Input 6:**

**3 6**

**1 4 5 2 3 2 5 4**

**Output 6:**

**-1**

**Input 7:**

**7 4**

**1 4 5 2 6 3 2 5**

**Output 7:**

**1**

**Input 8:**

**7 41**

**1 4 5 2 6 3 5 47 85 6**

**Output 8:**

**-1**

**Input 9:**

**3 2**

**1 4 5 2 63 254 15 7 58**

**Output 9:**

**3**

**Input 10:**

**6 3**

**14 25 36 52 154 75**

**Output 10:**

**-1**

**Solution:**

**n,k=map(int,(input().split()))**

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

**for i in l:**

**if(i==k):**

**print(l.index(i))**

**break**

**else:**

**print(-1)**

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

**Ques:27(589)**

**You are provided with an array in which all elements are repeated thrice except one which is repeated twice.Your task is to print that number.**

**O(n) time and O(1) extra space**

**Input Description:**

**First line contains a number denoting size of array ‘n’.Next line contains n space separated numbers**

**Output Description:**

**Print the number which is repeated twice**

**Sample Input :**

**5**

**13 12 13 12 13**

**Sample Output :**

**12**

**Test Cases:**

**Input 1:**

**7**

**4 1 2 6 3 3 2**

**Output 1:**

**2**

**Input 2:**

**6**

**1 1 1 2 2 3**

**Output 2:**

**2**

**Input 3:**

**8**

**4 1 5 2 2 2 1 6**

**Output 3:**

**1**

**Input 4:**

**6**

**4 2 2 2 1 4**

**Output 4:**

**4**

**Input 5:**

**3**

**1 2 2**

**Output 5:**

**2**

**Input 6:**

**9**

**4 4 4 5 5 5 1 2 1**

**Output 6:**

**1**

**Input 7:**

**6**

**12 15 42 12 35 62**

**Output 7:**

**12**

**Input 8:**

**7**

**14 51 26 63 35 48 26**

**Output 8:**

**26**

**Input 9:**

**5**

**100 200 200 100 100**

**Output 9:**

**200**

**Input 10:**

**6**

**1 4 5 2 6 5**

**Output 10:**

**5**

**Solution:**

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

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

**for i in l:**

**if(l.count(i)==2):**

**print(i)**

**break**

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

**Ques:28(507)**

**You are an intern at GUVI and the company wants to organise its data and delete unnecessary extra storage elements used. You are given k arrays of unequal dimensions. Sort the k arrays individually and concatenate them.**

**Input Description:**

**First line contains the number of arrays. Subsequent lines contain the size of the array followed by the elements of the array.**

**Output Description:**

**An array containing the sorted elements of k sorted arrays**

**Sample Input :**

**3**

**2**

**98 12**

**6**

**1 2 3 8 5 9**

**1**

**11**

**Sample Output :**

**12 98 1 2 3 5 8 9 11**

**Solution:**

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

**a=list(map(int,(input().split())))**

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

**l=[]**

**for i in range(0, (n - k + 1)):**

**flag = False**

**for j in range(0, k):**

**if (a[i + j] < 0):**

**l.append(a[i+j])**

**flag = True**

**break**

**if (not(flag)):**

**l.append('0')**

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

**Ques:34(586)**

**You are given with an array .Your task is calculate the difference between two consecutive number. And if difference is greater than ‘k’, print 1 else print 0**

**Input Description:**

**You are given two numbers ‘n’, ’m’. Next line contains n space separated integers.**

**Output Description:**

**Print 1 if the difference is greater than ‘m’.**

**Sample Input :**

**5 15**

**50 65 85 98 35**

**Sample Output :**

**1 1 0 0 0**

**Test Cases:**

**Input 1:**

**3 6**

**4 2 5**

**Output 1:**

**0 0 0**

**Input 2:**

**5 5**

**4 2 5 7 8**

**Output 2:**

**0 1 0 0 0**

**Input 3:**

**6 2**

**4 7 5 1 2 3**

**Output 3:**

**1 0 0 0 0 0**

**Input 4:**

**3 3**

**4 7 5**

**Output 4:**

**0 0 0**

**Input 5:**

**5 8**

**4 7 5 2 1**

**Output 5:**

**0 0 0 0 0**

**Input 6:**

**6 4**

**7 4 1 2 3 4**

**Output 6:**

**0 0 0 0 0 0**

**Input 7:**

**5 5**

**7 4 2 5 6**

**Output 7:**

**0 0 0 0 0**

**Input 8:**

**6 6**

**7 4 8 1 2 3**

**Output 8:**

**0 0 0 0 0 0**

**Input 9:**

**4 3**

**7 4 7 5**

**Output 9:**

**0 0 0 0**

**Input 10:**

**5 5**

**4 1 4 2 3**

**Output 10:**

**0 0 0 0 0**

**Solution:**

**n,m=map(int,(input().split()))**

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

**s=[]**

**for i in range(0,n):**

**for j in range(i+1,n):**

**d=l[j]-l[i]**

**if(d>m):**

**s.append('1')**

**break**

**else:**

**s.append('0')**

**print(\*s)**

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

**Ques:29(479)**

**Shreya is a brilliant girl. She likes to memorize the numbers. These numbers will be shown to her. As an examiner develop an algorithm to test her memory.**

**CONSTRAINTS**

**1<=Y,N,T<=1000**

**Input Description:**

**First line contains no. of test cases(Y). Next line contains a number N. Next line contains n space separated numbers Next line contains a number T denoting the number of questions asked from you regarding the given array. Next line contains T space separated numbers.**

**Output Description:**

**Print the occurrence of each number if present else “NOT PRESENT”**

**Sample Input :**

**10**

**1 1 1 2 2 2 3 8 9 7**

**5**

**1 2 3 0 5**

**Sample Output :**

**3 3 1 Not Present Not Present**

**Test Cases:**

**Input 1:**

**10**

**1 1 1 2 2 2 3 8 9 7**

**5**

**1 2 3 0 5**

**Output 1:**

**3 3 1 Not Present Not Present**

**Input 2:**

**5**

**1 1 4 2 6**

**3**

**1 42 2**

**Output 2:**

**2 Not Present 1**

**Input 3:**

**6**

**4 5 7 1 6 2**

**4**

**5 7 2 6**

**Output 3:**

**1 1 1 1**

**Input 4:**

**8**

**4 1 5 2 6 3 7 20**

**5**

**4 7 5 1 6**

**Output 4:**

**1 1 1 1 1**

**Input 5:**

**7**

**12 15 41 62 32 10 14**

**5**

**10 20 30 40 50**

**Output 5:**

**1 Not Present Not Present Not Present Not Present**

**Input 6:**

**5**

**1 2 4 1 2**

**3**

**1 4 5**

**Output 6:**

**2 1 Not Present**

**Input 7:**

**6**

**4 5 1 2 6 3**

**4**

**5 7 4 2**

**Output 7:**

**1 Not Present 1 1**

**Input 8:**

**6**

**4 1 5 2 3 4**

**5**

**7 4 2 6 3**

**Output 8:**

**Not Present 2 1 Not Present 1**

**Input 9:**

**2**

**14 15**

**3**

**51 14 26**

**Output 9:**

**Not Present 1 Not Present**

**Input 10:**

**5**

**74 84 52 63 21**

**6**

**14 15 20 31 24 74**

**Output 10:**

**Not Present Not Present Not Present Not Present Not Present 1**

**Solution:**

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

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

**n1=int(input())**

**l1=list(map(int,input().split()))**

**l2=[]**

**for i in l:**

**for j in l1:**

**if j in l:**

**l2.append(l.count(j))**

**else:**

**l2.append('Not Present')**

**break**

**print(\*l2)**

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

**Ques:30(586)**

**You are given with an array .Your task is calculate the difference between two consecutive number. And if difference is greater than ‘k’, print 1 else print 0**

**Input Description:**

**You are given two numbers ‘n’, ’m’. Next line contains n space separated integers.**

**Output Description:**

**Print 1 if the difference is greater than ‘m’.**

**Sample Input :**

**5 15**

**50 65 85 98 35**

**Sample Output :**

**0 1 0 0 0**

**Solution:**

**n,m=map(int,(input().split()))**

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

**s=[]**

**for i in range(0,n):**

**for j in range(i+1,n):**

**d=l[j]-l[i]**

**if(d>m):**

**s.append('1')**

**break**

**else:**

**s.append('0')**

**print(\*s)**

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

**Ques:31(593)**

**You are given an array. Your task is to sort the array in given manner. Print the elements in increasing order of the frequency. If frequency is same print smaller one first.**

**Input Description:**

**You are given a number ‘n’. Then in next line n space separated numbers.**

**Output Description:**

**Print the array as mentioned**

**Sample Input :**

**4**

**1 1 3 2**

**Sample Output :**

**2 3 1**

**Test Cases:**

**Input 1:**

**8**

**2 1 4 5 2 6 3 2**

**Output 1:**

**3 6 5 4 1 2**

**Input 2:**

**6**

**3 2 1 4 5 2**

**Output 2:**

**5 4 1 2 3**

**Input 3:**

**5**

**4 1 2 1 2**

**Output 3:**

**2 1 4**

**Input 4:**

**9**

**4 1 5 2 6 3 2 1 1**

**Output 4:**

**3 6 2 5 1 4**

**Input 5:**

**4**

**7 4 1 2**

**Output 5:**

**2 1 4 7**

**Input 6:**

**8**

**4 7 1 5 2 6 3 2**

**Output 6:**

**3 6 2 5 1 7 4**

**Input 7:**

**3**

**1 4 5**

**Output 7:**

**5 4 1**

**Input 8:**

**6**

**1 1 2 4 1 1**

**Output 8:**

**4 2 1**

**Input 9:**

**5**

**4 1 2 3 5**

**Output 9:**

**5 3 2 1 4**

**Input 10:**

**6**

**12 14 51 21 14 12**

**Output 10:**

**21 51 14 12**

**Solution:**

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

**s=list(map(int,(input().split())))**

**l=[]**

**for i in s:**

**if i not in l:**

**l.append(i)**

**print(\*l[::-1])**

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

**Ques:32(508)**

Ria is a 5 year old girl. Her mother wants to teach her how to sort words in the same order that they appear in a dictionary. She decides to write a program to sort a given set of strings based on their alphabetical order. Help Ria’s mother to complete the program.

Input Description:

A set of N strings

Output Description:

Alphabetically sorted set of strings

Sample Input :

3

InfinityWar EndGame Avengers

Sample Output :

Avengers EndGame InfinityWar

Test Cases:

Input 1:

5

python was developed by rossum

Output 1:

by developed python rossum was

Input 2:

4

live life to peaks

Output 2:

life live peaks to

Input 3:

3

get well soon

Output 3:

get soon well

Input 4:

4

speed thrills but kills

Output 4:

but kills speed thrills

Input 5:

5

coding is a lifetime skill

Output 5:

a coding is lifetime skill

Input 6:

3

i am mad

Output 6:

am i mad

Input 7:

5

fun is to have fun

Output 7:

fun fun have is to

Input 8:

5

love to code till end

Output 8:

code end love till to

Input 9:

4

hello i am python

Output 9:

am hello i python

Input 10:

1

hi

Output 10:

hi

Solution:

import re

n=int(input())

l=input()

a=re.findall(r'\w+',l)

d=list(map(str,a))

d.sort()

print(\*d)

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

Ques:33(505)

Loki wants to steal the tesseract but in order to do so, he has to rearrange the elements in an array in a specific manner which is mentioned in a clue. The clue says ‘cursed are the odd and sorted are the even’. Loki manages to decode the clue which translates to “sort the alternate elements of an array, starting from the element at index 0, in ascending order”. Manipulate the array so as to help Loki steal the tesseract.

Input Description:

Size of the array followed by the elements of the array

Output Description:

Array with alternate elements sorted in ascending order

Sample Input :

5

3 9 1 44 6

Sample Output :

1 9 3 44 6

Test Cases:

Input 1:

10

14 5 6 2 7 8 9 2 1 0

Output 1:

1 0 6 2 7 2 9 5 14 8

Input 2:

8

1 4 5 2 1 6 3 7

Output 2:

1 2 1 4 3 6 5 7

Input 3:

6

45 51 26 30 74 85

Output 3:

26 30 45 51 74 85

Input 4:

3

14 15 20

Output 4:

14 15 20

Input 5:

5

41 21 52 63 22

Output 5:

22 21 41 63 52

Input 6:

6

14 12 51 26 32 10

Output 6:

14 10 32 12 51 26

Input 7:

7

1 5 4 2 6 7 3

Output 7:

1 2 3 5 4 7 6

Input 8:

6

4 5 1 2 6 3

Output 8:

1 2 4 3 6 5

Input 9:

5

3 9 1 44 6

Output 9:

1 9 3 44 6

Input 10:

15

1 4 2 5 6 3 2 1 7 8 10 12 15 14 16

Output 10:

1 1 2 3 2 4 6 5 7 8 10 12 15 14 16

Solution:

n=int(input())

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

a=[]

b=[]

for i in range(len(l)):

    if(i%2==0):

        a.append(l[i])

    else:

        b.append(l[i])

c=[]

a.sort(reverse=False)

b.sort()

j=0

for j in range(len(b)):

    c.append(a[j])

    c.append(b[j])

    j=j+1

if(len(a)>len(b)):

    c.append(a[-1])

print(\*c)

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

Ques:34(611)

You are given with a task to print whether array is ‘majestic’ or not. A ‘majestic’ array is an array whose sum of first three number is equal to last three number.

Input Description:

You are given a number ‘n’,Next line contains ‘n’ space separated

Output Description:

Print 1 if array is majestic and 0 if it is not

Sample Input :

7

1 2 3 4 6 0 0

Sample Output :

1

Test Cases:

Input 1:

7

1 2 3 4 6 0 0

Output 1:

1

Input 2:

6

1 2 1 4 3 2

Output 2:

0

Input 3:

8

-1 2 2 3 2 3 0 0

Output 3:

1

Input 4:

5

7 4 1 2 3

Output 4:

0

Input 5:

7

1 4 2 3 5 1 1

Output 5:

1

Input 6:

9

1 4 5 2 6 3 2 1 7

Output 6:

1

Input 7:

8

4 1 5 2 6 3 1 1

Output 7:

0

Input 8:

6

1 1 1 1 1 1

Output 8:

1

Input 9:

7

1 2 1 0 2 2 0

Output 9:

1

Input 10:

8

4 1 5 2 6 3 2 1 4

Output 10:

0

Solution:

n=int(input())

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

s1=l[0]+l[1]+l[2]

s2=l[-1]+l[-2]+l[-3]

if(s1==s2):

    print('1')

else:

    print('0')

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

**Ques:35**

You are given an array of ids of prisoners. The jail authority found that there are some prisoners of same id. Your task is to help the authority in finding the common ids.

Input Description:

First line contains a number ‘n’ representing no of prisoners. Next line contains n space separated numbers.

Output Description:

Print the ids which are not unique. Print -1 if all ids are unique

Sample Input :

7

1 1 11 121 131 141 98

Sample Output :

1

Test Cases:

Input 1:

8

4 1 5 2 6 3 2 1

Output 1:

1

Input 2:

6

4 4 2 6 3 1

Output 2:

4

Input 3:

5

11 11 12 142 121

Output 3:

11

Input 4:

9

74 52 16 32 48 74 16 10 20

Output 4:

74

Input 5:

5

2 2 4 6 8

Output 5:

2

Input 6:

6

1 11 121 131 98 1

Output 6:

1

Input 7:

5

2 4 6 8 3

Output 7:

-1

Input 8:

3

1 11 121

Output 8:

-1

Input 9:

5

10 15 20 30 40

Output 9:

-1

Input 10:

6

1 12 24 35 36 42

Output 10:

-1

Solution:

n=int(input())

s=list(map(int,(input().split())))

for i in s:

    if(s.count(i)>1):

        print(i)

        break

else:

    print(-1)

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

Ques:36(600)

You are given two arrays of equal length. Your task is to merge the two arrays and then sort them too and then find the sum of two middlemost elements.

Input Description:

You are given a number ‘n’. Then Next line contains n separated numbers.Third line contains n space separated numbers.

Output Description:

Print the sum of two middle elements

Sample Input :

5

1 9 16 25 46

2 3 4 5 6

Sample Output :

11

Test Cases:

Input 1:

4

4 1 2 5

7 4 5 2

Output 1:

8

Input 2:

5

1 2 4 5 6

3 1 1 2 5

Output 2:

5

Input 3:

9

4 7 1 5 2 6 3 1 0

4 1 2 6 3 5 4 1 0

Output 3:

6

Input 4:

6

4 1 2 1 2 3

6 2 4 1 3 2

Output 4:

4

Input 5:

3

4 1 2

3 4 1

Output 5:

5

Input 6:

6

4 1 2 5 6 3

7 1 5 2 6 3

Output 6:

7

Input 7:

4

1 4 5 2

6 2 3 4

Output 7:

7

Input 8:

3

4 7 5

6 2 1

Output 8:

9

Input 9:

4

4 7 5 1

2 3 5 1

Output 9:

7

Input 10:

6

4 1 2 0 3 4

7 4 1 2 3 0

Output 10:

5

Solution:

n=int(input())

l1=list(map(int,(input().split())))

l2=list(map(int,(input().split())))

l=l1+l2

l.sort()

c=len(l)//2

a=l[c]

b=l[c-1]

s=a+b

print(s)

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

Ques:37(27)

Given 2 numbers N and K followed by N elements,print the number of repetition of K otherwise print '-1' if the element not found.

Sample Testcase :

INPUT

6 2

1 2 3 5 7 8

OUTPUT

0

Test Cases:

Input 1:

6 1

4 1 2 3 1 5

Output 1:

1

Input 2:

5 0

1 0 0 0 0

Output 2:

3

Input 3:

6 2

4 1 2 3 2 3

Output 3:

1

Input 4:

7 2

4 1 2 5 2 3 2

Output 4:

2

Input 5:

7 5

1 4 5 2 6 5 3

Output 5:

1

Input 6:

7 0

1 0 2 0 3 0 4

Output 6:

2

Input 7:

9 1

1 1 1 1 1 2 5 4 3

Output 7:

4

Input 8:

6 4

1 4 5 2 3 4

Output 8:

1

Input 9:

5 3

1 3 3 4 2

Output 9:

1

Input 10:

7 2

4 1 2 2 2 3 4

Output 10:

2

Solution:

n,k=map(int,(input().split()))

s=input().split()

b=list(s)

if b.count(str(k))==0:

    print('-1')

else:

    print(int(b.count(str(k))-1))

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

Ques:38(395)

Prakash is bored and wants to spends his time. He starts rolling a die and observes the value that is shown. He rolls the dice N times and observes that just one number appears distinctly, all the others get repeated or does not appear at all. Find out which was the number that puzzled Prakash for sometime, after which he realised that it was just a coincidence.

Constraints

0 <   N  <= 100

0 <= A[i] <= 100

Input Description:

The first line contains a positive integer N, denoting the size of the array. The second line contains N positive integers, denoting the face values that appeared when the die was rolled.

Output Description:

Print out the singly occurring number.

Sample Input :

5

1 1 2 5 5

Sample Output :

2

Test Cases:

Input 1:

5

1 4 5 1 2

Output 1:

4

5

2

Input 2:

10

7 4 5 8 1 5 4 7 8 2

Output 2:

1

2

Input 3:

9

4 -1 -2 -1 -2 3 4 0 5

Output 3:

3

0

5

Input 4:

9

4 1 2 3 5 4 1 2 3

Output 4:

5

Input 5:

8

4 7 5 1 4 7 5 2

Output 5:

1

2

Input 6:

3

1 1 2

Output 6:

2

Input 7:

5

1 1 4 2 3

Output 7:

4

2

3

Input 8:

6

1 4 2 3 5 4

Output 8:

1

2

3

5

Input 9:

5

1 1 2 5 5

Output 9:

2

Input 10:

2

1 2

Output 10:

1 2

Solution:

n=int(input())

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

for i in l:

    if(l.count(i)==1):

        print(i)

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

Ques:39(212)

Given a number N and an array of N integers, find the sum of all the negative numbers in the array.

Input Size : N <= 100000

Sample Testcase :

INPUT

2

3 0

OUTPUT

0

Test Cases:

Input 1:

2

-122 -144

Output 1:

-266

Input 2:

3

-1 -5 -7

Output 2:

-13

Input 3:

6

1 2 -4 -5 -12 2

Output 3:

-21

Input 4:

7

4 -5 -6 -8 -9 -1 2

Output 4:

-29

Input 5:

6

4 1 5 2 0 -3

Output 5:

-3

Input 6:

9

1 2 4 5 7 8 2 3 -5

Output 6:

-5

Input 7:

8

4 0 1 2 0 -5 -2 3

Output 7:

-7

Input 8:

6

1 2 -3 -4 -8 2

Output 8:

-15

Input 9:

5

1 2 -1 -2 3

Output 9:

-3

Input 10:

1

-8

Output 10:

-8

Solution:

n=int(input())

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

s=[]

for i in l:

    if(i<0):

        s.append(i)

print(sum(s))

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

Ques:40(474)

Rajesh and Ram are having a conflict on the maximum marks that they have scored in all the exams conducted in the past year. The one having scored the maximum gets a treat from the other. They decide to go through their test papers and record their highest marks. You are Rajesh’s best friend and as he has tutions to attend, he gives you all his test papers and asks you to find out the maximum marks that he has scored among all the marks in all exams. He promises you a treat if he wins the bet with Ram. Help Rajesh find out his highest marks.

Constraints:

1 <= N <= 10

0 <= A[] <= 100

Input Description:

First line contains count of marks. Next line is the list of marks obtained by Rajesh.

Output Description:

Highest marks obtained by Rajesh.

Sample Input :

3

82 96 72

Sample Output :

96

Test Cases:

Input 1:

7

47 58 96 62 54 51 53

Output 1:

96

Input 2:

6

45 15 26 34 51 85

Output 2:

85

Input 3:

5

98 94 54 74 52

Output 3:

98

Input 4:

4

92 98 94 97

Output 4:

98

Input 5:

2

99 98

Output 5:

99

Input 6:

10

45 75 85 96 62 32 99 98 94 75

Output 6:

99

Input 7:

9

47 51 52 63 24 84 51 60 30

Output 7:

84

Input 8:

7

41 51 45 63 25 74 85

Output 8:

85

Input 9:

5

41 52 63 23 21

Output 9:

63

Input 10:

3

10 20 30

Output 10:

30

Solution:

n=int(input())

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

print(max(l))

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

Ques:41(490)

Ria is always fascinated by the number 2. She always wants to know who came second in a race, the second person to set foot on the moon and so on. She is given a list of numbers and asked to find the maximum. As always, she reports the second highest number as the maximum because according to her, 2 is higher than 1. Find out which was the number that Ria would have reported, given a list of N numbers.

Input Description:

Size of the array followed by the elements of the array.

Output Description:

Second largest element of the array.

Sample Input :

10

1 9 8 7 6 5 2 3 4 10

Sample Output :

9

Test Cases:

Input 1:

8

47 45 85 96 62 33 21 24

Output 1:

85

Input 2:

5

120 124 121 130 100

Output 2:

124

Input 3:

3

124 152 232

Output 3:

152

Input 4:

4

12 21 32 24

Output 4:

24

Input 5:

6

41 52 23 21 24 12

Output 5:

41

Input 6:

7

4 7 5 8 9 6 2

Output 6:

8

Input 7:

3

1 4 2

Output 7:

2

Input 8:

6

4 1 5 2 3 9

Output 8:

5

Input 9:

10

4 7 8 5 9 6 52 31 52 20

Output 9:

52

Input 10:

5

4 6 3 2 8

Output 10:

6

Solution:

n=int(input())

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

l.sort()

print(l[-2])

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

topic:strings

Ques:42(9)

Given a string S, print it after changing the middle element to \* (if the length of the string is even, change the 2 middle elements to \*).

Sample Testcase :

INPUT

hello

OUTPUT

he\*lo

Test Cases:

Input 1:

live

Output 1:

l\*\*e

Input 2:

smile

Output 2:

sm\*le

Input 3:

apple

Output 3:

ap\*le

Input 4:

coding

Output 4:

co\*\*ng

Input 5:

happy

Output 5:

ha\*py

Input 6:

guido van rossum

Output 6:

guido v\*\* rossum

Input 7:

program

Output 7:

pro\*ram

Input 8:

world

Output 8:

wo\*ld

Input 9:

python

Output 9:

py\*\*on

Input 10:

talentio

Output 10:

tal\*\*tio

Solution:

a=input()

s=list(a)

l=len(s)

if(l%2==0):

l=l//2

s[l]="\*"

s[l-1]="\*"

else:

l=int(l/2)

s[l]="\*"

print(''.join(s),end="")

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

Topic: strings

Ques:43(39)

Given a sentence and string S, find how many times S occurs in the given sentence.If S is not found in the sentence print -1

Input Size : |sentence| <= 1000000(complexity O(n)).

Sample Testcase :

INPUT

I enjoy coding

coding

OUTPUT

1

Test Cases:

Input 1:

hill will bill

hill

Output 1:

1

Input 2:

cat rat bat cat cat

cat

Output 2:

3

Input 3:

coding is a skill and is fun

is

Output 3:

2

Input 4:

live long live happy live healthy

live

Output 4:

3

Input 5:

the one who knows everything are the one who forget everything

one

Output 5:

2

Input 6:

abc def ghi abc defghi

def

Output 6:

1

Input 7:

tom and jerry always fight but jerry always wins

jerry

Output 7:

2

Input 8:

jack and jill went to the well and fell

and

Output 8:

2

Input 9:

python is easy to learn and python is a scripting language

python

Output 9:

2

Input 10:

i love python

coding

Output 10:

-1

Solution:

l=input()

s=input()

a=l.split(' ')

if a.count(s)== 0:

    print('-1')

else:

    print(a.count(s))

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

Ques:44(492)

The prison warden at Central jail is given a tip-off that a prison inmate is planning an escape. The warden suspects a particular prisoner of planning an escape and wants to find out if he/she is present in his/her cell. The layout of the prison is modelled in a matrix with every cell of the matrix representing a prison cell. The matrix is filled with the prisoner ids at the corresponding cells. Find out whether the person the warden suspects is present in the prison or not.

Input Description:

First line contains the dimensions of the prison matrix, followed by the ids of prisoners as elements of the matrix. The third line contains the id to be searched.

Output Description:

(yes/no) whether the given element is present in the matrix or not.

Sample Input :

2 5

2 3 0 7 1 5 3 4 1 8

11

Sample Output :

No

Test Cases:

Input 1:

3 2

4 7 15 26 32 21

10

Output 1:

no

Input 2:

5 1

41 21 35 62 30

35

Output 2:

yes

Input 3:

4 1

45 26 32 21

32

Output 3:

yes

Input 4:

3 1

41 52 63

52

Output 4:

yes

Input 5:

2 4

4 7 8 5 9 3 2 1

5

Output 5:

yes

Input 6:

2 3

4 1 5 2 6 3

1

Output 6:

yes

Input 7:

3 3

4 1 22 35 15 42 63 84 20

1

Output 7:

yes

Input 8:

6 3

4 5 7 1 2 6 3 8 7 4 5 9 2 1 11 23 54 45

5

Output 8:

yes

Input 9:

2 2

4 7 5 1

3

Output 9:

no

Input 10:

3 2

4 5 7 8 4 1

6

Output 10:

no

Solution:

n,k=map(int,(input().split()))

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

ids=int(input())

for i in l:

    if(i==ids):

        print('yes')

        break

else:

    print('no')

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

Ques:46(596)

You are given an array of digits. Your task is to print the digit with maximum frequency.

Input Description:

You are given length of array ’n’,next line contains n space separated numbers.

Output Description:

Print the number with maximum frequency. If two numbers have equal frequency print the number that comes first

Sample Input :

7

1 2 3 4 4 4 5

Sample Output :

4

Test Cases:

Input 1:

5

41 26 32 21 41

Output 1:

41

Input 2:

3

31 31 22

Output 2:

31

Input 3:

5

12 24 12 23 25

Output 3:

12

Input 4:

9

7 4 1 5 2 1 5 4 3

Output 4:

4

Input 5:

8

4 4 4 1 2 5 3 2

Output 5:

4

Input 6:

7

4 1 5 2 6 3 3

Output 6:

3

Input 7:

8

4 1 2 0 2 3 5 3

Output 7:

2

Input 8:

6

4 1 2 0 3 2

Output 8:

2

Input 9:

4

7 4 1 1

Output 9:

1

Input 10:

7

4 5 1 2 4 5 5

Output 10:

4

Solution:

n=int(input())

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

for i in l:

    if(l.count(i)!=1):

        print(i)

        break

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

Ques:47(591)

Ramit is given a list of both positive and negative integers. He has to tell the maximum sum out of all sub arrays in the given list. He got confused and requested help from you. Now it is your task to find the maximum sum out of all sub arrays in the given list.

Input Description:

You are given a number 'n'. Next line contains n space separated numbers.

Output Description:

Print the max sum of subarray.

Sample Input :

5

1 2 3 -2 5

Sample Output :

9

Test Cases:

Input 1:

5

-1 -4 -2 -3 -1

Output 1:

-1

Input 2:

6

-4 -5 -2 1 3 6

Output 2:

10

Input 3:

9

1 4 2 1 5 3 -2 -1 -4

Output 3:

16

Input 4:

7

4 1 -2 -1 -3 6 5

Output 4:

11

Input 5:

3

1 2 4

Output 5:

11

Input 6:

6

4 1 2 -4 -5 3

Output 6:

7

Input 7:

7

4 1 2 5 -8 -5 3

Output 7:

12

Input 8:

4

3 2 1 4

Output 8:

10

Input 9:

6

4 7 1 2 1 3

Output 9:

18

Input 10:

5

1 2 3 -2 5

Output 10:

9

Solution:

n=int(input())

a=list(map(int,input().split()))

sublist=[]

for i in range(len(a)+1):

    for j in range(i+1,len(a)+1):

        s=a[i:j]

        sublist.append(sum(s))

print(max(sublist))

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

Ques:49(303)

Given a number N, print the sum of squares of all its digits.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

12

OUTPUT

5

Test Cases:

Input 1:

120

Output 1:

5

Input 2:

47

Output 2:

65

Input 3:

23

Output 3:

13

Input 4:

11

Output 4:

2

Input 5:

62

Output 5:

40

Input 6:

4215

Output 6:

46

Input 7:

6231

Output 7:

50

Input 8:

12

Output 8:

5

Input 9:

523

Output 9:

38

Input 10:

154

Output 10:32

Solution:

n=int(input())

s=0

while(n>0):

    r=n%10

    s=s+r\*\*2

    n=n//10

print(s)

Ques:50(45)

Given 2 numbers N,K followed by N elements print all the elements lesser than K in sorted order.If the elements could not be found print -1

Input Size : N <= 100000

Sample Test Case :

INPUT

5 3

1 2 1 4 1

OUTPUT

1 1 1 2

Test Cases:

Input 1:2 3

1 2

Output 1:1 2

Input 2:4 2

1 3 2 5

Output 2:1

Input 3:3 1

0 7 2

Output 3:0

Input 4:2 5

3 7

Output 4:3

Input 5:2 10

2 8

Output 5:-1

Input 6:4 8

1 9 7 2

Output 6:1 2 7

Input 7:

5 11

11 10 101 3 4

Output 7:

3 4 10

Input 8:

2 2

2 2

Output 8:

-1

Input 9:

3 6

6 6 5

Output 9:

5

Input 10:

5 5

1 1 1 1 1

Output 10:

1 1 1 1 1

Solution:

n,k=map(int,(input().split()))

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

l.sort()

s=[]

for i in l:

    if(i<k):

        s.append(i)

if(len(s)==0):

    print(-1)

else:

    print(\*s)

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

Ques:51(601)

Rajesh was going through alternative array sorting. He wishes to print the array alternatively. Hence hired you. Your task is to help rajesh in printing the array alternatively.

An alternative array is an array in which the first element is maximum of the whole array, the second element is minimum of the whole array.Third element is second largest. Fourth element is the second smallest And so on.print the array in the desired manner.

Input Description:

You are given with the length of array ‘n’. followed by ‘n’ space separated numbers.

Output Description:

Print the array as mentioned.

Sample Input :

5 1 7 11 16 19

Sample Output :

19 1 16 7 11

Test Cases:

Input 1:

4 5 12 74 85 63 54 23

Output 1:

23 5 54 12 63 74 85

Input 2:

1 4 5 2 6 3 7

Output 2:

7 4 3 5 6 2

Input 3:

2 5 4 7 8 9 6

Output 3:

6 5 9 4 8 7

Input 4:

1 1 2 4 5 3

Output 4:

3 1 5 2 4

Input 5:

4 5 1 0 2

Output 5:

2 5 0 1

Input 6:

11 24 15 63 20 45

Output 6:

45 24 20 15 63

Input 7:

74 51 56 32 41 20

Output 7:

20 51 41 56 32

Input 8:

120 145 163 157 182 235

Output 8:

235 145 182 163 157

Input 9:

45 17 52 63 54 95 63

Output 9:

63 17 95 52 54 63

Input 10:

1 2 4 2 1 3 6

Output 10:

6 2 3 4 1 2

Solution:

a=list(map(int,input().split()))

A=a[1:]

n=len(A)

i=0

j=n-1

while(i<j):

    print(A[j],end=" ")

    j=j-1

    print(A[i],end=" ")

    i=i+1

if(n%2!=0):

print(A[i])

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

Ques:52(494)

Ram is the CEO of an MNC. He wants to order the employee salaries in ascending order so that he can do a salary hike based on the salary values of employees. He selects you to do the task of sorting the salaries. Sort the salaries in ascending order and pass on the information to Ram.

Input Description:

Number of employees followed by the salaries of employees.

Output Description:

Salaries sorted in ascending order.

Sample Input :

8

7000 8000 6500 1200 4000 2800 3000 5230

Sample Output :

1200 2800 3000 4000 5230 6500 7000 8000

Test Cases:

Input 1:

4

5000 3000 1000 4000

Output 1:

1000 3000 4000 5000

Input 2:

3

2300 4500 1010

Output 2:

1010 2300 4500

Input 3:

6

6000 2000 4000 3500 2800 1000

Output 3:

1000 2000 2800 3500 4000 6000

Input 4:

4

2800 5400 1230 3200

Output 4:

1230 2800 3200 5400

Input 5:

2

7000 1000

Output 5:

1000 7000

Input 6:

1

1000

Output 6:

1000

Input 7:

8

1200 9000 3000 2300 5400 5700 2300 4560

Output 7:

1200 2300 3000 4560 5400 5700 9000 2300

Input 8:

4

9000 8700 3400 3720

Output 8:

3400 3720 8700 9000

Input 9:

0

12

Output 9:12

Input 10:

4

7000 8000 3200 6700

Output 10:

3200 6700 7000 8000

Solution:

n=int(input())

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

l.sort()

print(\*l)

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

Ques:53(17)

Given a string S, print 2 strings such that first string containing all characters in odd position(s) and other containing all characters in even position(s).

Sample Testcase :

INPUT

XCODE

OUTPUT

XOE CD

Test Cases:

Input 1:

disney

Output 1:

dse iny

Input 2:

target

Output 2:

tre agt

Input 3:

infytq

Output 3:

ift nyq

Input 4:

script

Output 4:

srp cit

Input 5:

ladder

Output 5:

lde adr

Input 6:

flower

Output 6:

foe lwr

Input 7:

programmer

Output 7:

porme rgamr

Input 8:

wise

Output 8:

Input 9:

hello

Output 9:

hlo el

Input 10:

talentio

Output 10:

tlni aeto4rr

Solution:

s=input()

l1=s[::2]

l2=s[1::2]

print(l1,end=" ")

print(l2)

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

Ques:54(44)

Given a number N and 2 arrays A and B of sorted order of size N, print the common elements.If it is not found print -1.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

5

1 1 1 1 1

1 2 3 4 5

OUTPUT

1

Test Cases:

Input 1:

4

1 2 4 3

8 2 7 1

Output 1:

1 2

Input 2:

3

9 4 8

9 8 7

Output 2:

9 8

Input 3:

2

3 6 1

9 7 6

Output 3:

6

Input 4:

5

1 6 5 3

7 5 4 3

Output 4:

3 5

Input 5:

3

2 5

8 7

Output 5:

-1

Input 6:

2

5 6

7 5

Output 6:

5

Input 7:

5

1 2 3 6 3

8 5 7 1 3

Output 7:

1 3

Input 8

2

1 6

1 6

Output 8:

1 6

Input 9:

3

1 8 3

1 2 3

Output 9:

1 3

Input 10:

3

0 1 10

3 4 6

Output 10:

-1

Solution:

n=int(input())

a=list(map(int,(input().split())))

b=list(map(int,(input().split())))

l=[]

for i in b:

    if i in a:

        l.append(i)

if(len(l)==0):

    print(-1)

else:

    print(\*l)

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

Ques:55(36)

Given 2 strings S1 and s2, check whether they are case senitively equal without using any predefined function(case sensitive).If they are not same print 'no'

Sample Testcase :

INPUT

guvi guvi

OUTPUT

yes

Test Cases:

Input 1:111 111

Output 1:yes

Input 2:python Pytho

Output 2:no

Input 3:00 oo

Output 3:no

Input 4:123 123

Output 4:yes

Input 5:123 '123'

Output 5:no

Input 6:a\_a a\_a

Output 6:yes

Input 7:\_ \_\_

Output 7:no

Input 8:12AB 12ab

Output 8:no

Input 9:hey HEY

Output 9:no

Input 10:abcd dcba

Output 10:no

Solution:

s1,s2=map(str,(input().split()))

if(s1[::]==s2[::]):

  print('yes')

else:

  print('no')

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

Ques:56(49)

Given a number N and an array of N elements, find the length of the longest repeating sequence of the elements.If no such sequence is found print -1

Input Size : N <= 100000

Sample Testcase :

INPUT

8

1 2 2 2 3 4 5 6

OUTPUT

3

Test Cases:

Input 1:

6

1 4 2 5 6 6

Output 1:

2

Input 2:

5

1 4 7 4 4

Output 2:

3

Input 3:

5

1 4 4 5 2

Output 3:

2

Input 4:

7

4 5 7 4 4 4 2

Output 4:

4

Input 5:

6

1 2 2 2 3 6

Output 5:

3

Input 6:

6

1 2 2 2 2

Output 6:

4

Input 7:

4

1 4 4 2

Output 7:

2

Input 8:

6

1 4 5 2 3 3

Output 8:

2

Input 9:

5

1 4 5 2 3

Output 9:

-1

Input 10:

8

1 1 2 2 2 2 3 4

Output 10:

4

Solution:

n=int(input())

a=list(map(int,(input().split())))

l=[]

for i in a:

    if(a.count(i)==1):

        continue

    else:

        b=a.count(i)

        l.append(b)

if(len(l)==0):

    print(-1)

else:

    print(max(l))

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

Ques:57(80)

Given a string S, print the encoded string by adding 3 to each character(a maps to d,b maps to e,c maps to f and so on).

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

RADAR

OUTPUT

UDGDU

Test Cases:

Input 1:

SOUTH

Output 1:

VRXWK

Input 2:

NORTH

Output 2:

QRUWK

Input 3:

EAST

Output 3:

HDVW

Input 4:

TALENTIO

Output 4:

WDOHQWLR

Input 5:

parrot

Output 5:

Sduurw

Input 6:

sita

Output 6:

vlwd

Input 7:

sandhya12

Output 7:

Vdqgkbd45

Input 8:

sairam98

Output 8:

vdludp<;

Input 9:

123

Output 9:

456

Input 10:

0sun

Output 10:

3vxq

Solution:

s=input()

n=""

k=3

for i in range(len(s)):

    val=ord(s[i])

    dup=3

    if(val+3>122):

        k-=(122-val)

        k=k%26

        n+=chr(96+k)

    else:

n+=chr(val+k)

    k=dup

print(n)

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

Ques:58(58)

Given 2 array of size N and M.merge them in sorted order and print it.

Input Size : |N||M| <= 100000( O(n))

Sample Testcase :

INPUT

5 4

1 2 3 4 5

1 2 3 4

OUTPUT

1 1 2 2 3 3 4 4 5

Test Cases:

Input 1:

1 3 4

5 7 8

Output 1:

1 3 4 5 7 8

Input 2:

4

1 3 6 4

Output 2:

1 3 4 4 6

Input 3:

2 4 1

5 1 9

Output 3:

1 1 2 4 5 9

Input 4:

7 1 9

10 2

Output 4:

1 2 7 9 10

Input 5:

5 1 0

9 3 7

Output 5:

0 1 3 5 7 9

Input 6:

2

1 2

Output 6:

1 2 2

Input 7:

0 10

9 2 8

Output 7:

0 2 8 9 10

Input 8:

9

1 2

Output 8:

1 2 9

Input 9:

4 8 1

7 -1 5

Output 9:

-1 1 4 5 7 8

Input 10:

-1 2 8 1 9

4 5 6 3

Output 10:

-1 1 2 3 4 5 6 8 9

Solution:

a=list(map(int,(input().split())))

b=list(map(int,(input().split())))

l=[\*a,\*b]

l.sort()

print(\*l)

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

Ques:59(81)

Given 3 points check whether they lie on the same line.If they lie on the same line print 'yes' Otherwise print 'no'.

Sample Testcase :

INPUT

0 1

0 0

0 2

OUTPUT

Yes

Test Cases:

Input 1:

1 2

0 1

0 0

Output 1:

no

Input 2:

1 1

0 0

1 0

Output 2:

no

Input 3:

1 2

1 1

0 0

Output 3:

no

Input 4:

0 0

0 0

0 0

Output 4:

yes

Input 5:

1 1

1 1

1 1

Output 5:

yes

Input 6:

1 2

1 2

1 3

Output 6:

yes

Input 7:

1 0

0 1

1 3

Output 7:

no

Input 8:

3 4

5 6

7 8

Output 8:

yes

Input 9:

0 -1

-2 -4

-5 -7

Output 9:

no

Input 10:

1 1

2 2

3 3

Output 10:

yes

Solution:

x1,y1=map(int,(input().split()))

x2,y2=map(int,(input().split()))

x3,y3=map(int,(input().split()))

a=(y3-y2)\*(x2-x1)

b=(y2-y1)\*(x3-x2)

if(a==b):

    print('yes')

else:

    print('no')

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

Ques:60(486)

Your old mobile phone gets broken and so you want to purchase a new smartphone and decide to go through all the online websites to find out which dealer has the best offer for a particular model. You document the prices of N dealers. Dealer ids start from 0 and go up to N.  Find out which dealer has the best price for you.

Constraints:

1 <= N <= 100

1 <= A[] <= 100000

Input Description:

Number of dealers followed by the price offered by each dealer

Output Description:

Dealer offering the best price.

Sample Input :

3

10000 11200 12030

Sample Output :

Dealer0

Test Cases:

Input 1:

3

12000 40000 34000

Output 1:

dealer0

Input 2:

2

100000 12000

Output 2:

dealer1

Input 3:

1

12000

Output 3:

dealer0

Input 4:

4

3000 2000 10000 20000

Output 4:

dealer1

Input 5:

2

12000 3000

Output 5:

3000

Input 6:

3

10000 8700 45000

Output 6:

dealer1

Input 7:

1

10020

Output 7:

dealer0

Input 8:

2

9000 2000

Output 8:

dealer1

Input 9:

-3000 0

Output 9:

dealer0

Input 10:

2

1 2

Output 10:

dealer0

Solution:

n=int(input())

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

a=min(l)

print('Dealer',end="")

print(l.index(a))

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

Ques:61(67)

Given a string 'S' swap the even and odd characters starting from index 1(Assume the index starts from 0).

Input Size : |s| <= 10000000(complexity O(n))

Sample Testcase :

INPUT

codekata

OUTPUT

ocedakat

Test Cases:

Input 1:

codevita

Output 1:

ocedivat

Input 2:

talentio

Output 2:

ateltnoi

Input 3:

solution

Output 3:

osulitno

Input 4:

is

Output 4:

si

Input 5:

hand

Output 5:

ahdn

Input 6:

h

Output 6:

IndexError

Input 7:

overcome

Output 7:

voreocem

Input 8:

list

Output 8:

ilts

Input 9:

errors

Output 9:

reorsr

Input 10:

sita

Output 10:

isat

Solution:

s=input()

l=list(s)

for i in range(0,len(s),2):

    temp=l[i]

    l[i]=l[i+1]

    l[i+1]=temp

    s="".join(l)

print(s)

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

Ques:62(78)

Given a number N and an array of N strings, find the number of strings that are an anagram of 'kabali'.If there exists no anagram for the given string print '0'.

Input Size : 1 <= N <= 1000

Sample Testcase :

INPUT

5

kabali

kaabli

kababa

kab

kabail

OUTPUT

3

Test Cases:

Input 1:

3

cat

tac

act

Output 1:

3

Input 2:

4

Cat

Parrot

Suit

three

Output 2:

0

Input 3:

5

Three

Thr

Ee

Thr

thre

Output 3:

2

Input 4:

3

Python

Pyth

on

Output 4:

0

Input 5:

2

Py

p

Output 5:

0

Input 6:

7

Kitten

Kitt

Tten

Ten

Tten

It

ki

Output 6:

2

Input 7:

3

Super

Per

su

Output 7:

0

Input 8:

6

Sample

Sam

Samp

Sam

Sampl

e

Output 8:

2

Input 9:

4

Earth

Ear

Eart

ea

Output 9:

0

Input 10:

2

Star

st

Output 10:

0

Solution:

n=int(input())

l=[]

for i in range(n):

    a=input()

    s=sorted(a)

    l.append(s)

for j in l:

    a=l.count(j)

    if(a>1):

        print(a)

        break

else:

print(0)

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

Ques:63(608)

Prateek finds it difficult to judge the minimum element in the list of elements given to him. Your task is to develop the algorithm in order to find the minimum element.

Input Description:

You are given ‘n’ number of elements. Next line contains n space separated numbers.

Output Description:

Print the minimum element

Sample Input :

5

3 4 9 1 6

Sample Output :

1

Test Cases:

Input 1:

3

6 4 3

Output 1:

3

Input 2:

4

1 4 7 2

Output 2:

1

Input 3:

2

1 0

Output 3:

0

Input 4:

6

1 9 17 5 0 -1 -2

Output 4:

-2

Input 5:

2

-3 -5

Output 5:

-5

Input 6:

3

1 10 0

Output 6:

0

Input 7:

1

20

Output 7:

20

Input 8:

4

1 6 8 0

Output 8:

0

Input 9:

5

9 89 8 5 32

Output 9:

5

Input 10:

4

32 78 45 83

Output 10:

32

Solution:

n=int(input())

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

print(min(l))

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

Ques:64(86)

Given a sentence S take out the extra spaces.If no extra space is present print the same as output.

Input Size : |s| <= 100000(complexity O(n))

Sample Testcase :

INPUT

codekata   challenge

OUTPUT

codekata challenge

Test Cases:

Input 1

You  tube

Output 1:

You tube

Input 2:

Welcome  to  python

Output 2:

Welcome to python

Input 3:

Hello   world

Output 3:

Hello world

Input 4:

My   project

Output 4:

My proect

Input 5:

All    changes

Output 5:

All changes

Input 6:

Hello everyone

Output 6:

Hello everyone

Input 7:

Face    book

Output 7:

Face book

Input 8:

Programming   is   good

Output 8:

Programming is good

Input 9:

Hexa   decimal

Output 9:

Hexa decimal

Input 10:

Potato   chips

Output 10:

Potato chips

Solution:import re

s=input()

a=re.sub(' +', ' ', s)

print(a)

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

Ques:65(92)

Given 2 numbers N,K followed by a sorted array of N elements, search and tell if an element K is present in the array.print 'yes' if element is present otherwise print 'no'.

Input Size : 1 <= N <= 1000000000000000(Do it in logN time complexity)

Sample Testcase :

INPUT

3 2 7

 7

OUTPUT

yes

Test Cases:

Input 1:

1 2 3 4 5

3

Output 1:

yes

Input 2:

3 4 5 6

5

Output 2:

yes

Input 3:

6 7 8 90

90

Output 3:

yes

Input 4:

4 2 5 6

8

Output 4:

no

Input 5:

 8 9 0 4

5

Output 5:

no

Input 6:

9 7 5 4

8

Output 6:

no

Input 7:

1 2 3 4

1

Output 7:

yes

Input 8:

7 6 5 4

4

Output 8:

yes

Input 9:

8 6 5 3

6

Output 9:

yes

Input 10:

 6 3 2 1

2

Output 10:

yes

Solution:

x = input()

l = x.split()

k = input()

for e in l:

    if k == e:

        print("yes")

        break

else:

    print("no")

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

Ques:66(74)

Given a string S, print the reverse of the string after removing the vowels.If the resulting string is empty print '-1'.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codekata

OUTPUT

tkdc

Test Cases:

Input 1:

insert

Output 1:

trsn

Input 2:

changes

Output 2:

sgnhc

Input 3:

bingo

Output 3:

gnb

Input 4:

india

Output 4:

dn

Input 5:

steel

Output 5:

lts

Input 6:

program

Output 6:

mrgrp

Input 7:

normal

Output 7:

lmrn

Input 8:

youtube

Output 8:

bty

Input 9:

outline

Output 9:

nlt

Input 10:

sweets

Output 10:

stws

Solution:

s=input()

v=('a','e','i','o','u','A','E','I','O','U')

for i in s:

    if i in v:

        s=s.replace(i,"")

s1=s[::-1]

if(len(s1)==0):

    print('-1')

else:

    print(s1)

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

Ques:67(102)

Given a number N, followed by an array of N elements,print 'yes' if it is a sorted array otherwise print 'no'.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

3

2 3 7

OUTPUT

yes

Test Cases:

Input 1:

4

1 2 3 4

Output 1:

yes

Input 2:

5

5 6 5 4 7

Output 2:

no

Input 3:

4

5 6 4 6

Output 3:

no

Input 4:

4

 7 6 8 6

Output 4:

no

Input 5:

4

5 6 7 8

Output 5:

yes

Input 6:

4

7 8 9 10

Output 6:

yes

Input 7:

4

7 8 9 0

Output 7:

no

Input 8:

6

7 8 9 10 11 12

Output 8:

yes

Input 9:

3

1 2 3

Output 9:

yes

Input 10:

5

9  6 8 5 7

Output 10:

no

Solution:

n=int(input())

l=list(map(str,(input().split())))

if(n==0 or n==1):

    print('yes')

for i in range(1,n):

    if(l[i-1]>l[i]):

        print('no')

        break

else:

    print('yes')

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

Ques:68(34)

Given 2 numbers N and K followed by N elements, find the Kth smallest element.If the element cannot be found then print -1

Input Size : N <= 100000

Sample Testcase :

INPUT

5 2

1 1 2 4 5

OUTPUT

2

Test Cases:

Input 1:

3 4

1 2 5

Output 1:

-1

Input 2:

4 2

1 2 3 4

Output 2:

2

Input 3:

4 7

1 2 3 4

Output 3:

-1

Input 4:

3 9

1 9 6

Output 4:

9

Input 5:

3 7

1 2 7

Output 5:

7

Input 6:

5 11

1 2 3 4 5

Output 6:

-1

Input 7:

4 5

3 4 5 6 7

Output 7:

5

Input 8:

2 3

3 3

Output 8:

3

Input 9:

5 9

9 9 9 9 9

Output 9:

9

Input 10:

3 4

2 3 4

Output 10:

4

Solution:

n,k=map(int,(input().split()))

a=list(map(int,(input().split())))

l=[]

for i in a:

    if i not in l:

        l.append(i)

l.sort()

if(k<=len(l)):

    print(l[k-1])

else:

    print(-1)

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

Ques:70(120)

Given a number N in decimal convert it into binary value.

Input Size : N <= 100000

Sample Testcase :

INPUT

34

OUTPUT

100010

Test Cases:

Input 1:

9

Output 1:

1001

Input 2:

5

Output 2:

101

Input 3:

65

Output 3:

1000001

Input 4:

78

Output 4:

1001110

Input 5:

23

Output 5:

10111

Input 6:

45

Output 6:

101101

Input 7:

6

Output 7:

110

Input 8:

17

Output 8:

10001

Input 9:

3

Output 9:

11

Input 10:

87

Output 10:

1010111

Solution:

d=int(input())

b=bin(d)

print(b[2:])

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

Ques:71(135)

Given a number N and an array of N elements, print the array after removing duplicate elements.If no duplicate elements found print the same.

Input Size : N <= 100000

Sample Testcase :

INPUT

4

2 4 4 2

OUTPUT

2 4

Test Cases:

Input 1:

5

2 3 4 4 3

Output 1:

2 3 4

Input 2:

6

1 2 3 4 3 2

Output 2:

1 2 3 4

Input 3:

4

5 6 5 7

Output 3:

5 6 7

Input 4:

3

1 2 3

Output 4:

1 2 3

Input 5:

5

1 2 3 9 3 2

Output 5:

1 2 3 9

Input 6:

5

1 2 9 8 7

Output 6:

1 2 7 8 9

Input 7:

6

9 8 7 6 5 4

Output 7:

4 5 6 7 8 9

Input 8:

4

3 6 8 6

Output 8:

3 6 8

Input 9:

3

6 6 6

Output 9:

6

Input 10:

4

8 9 8 9

Output 10:

8 9

Solution:

n=int(input())

a=list(map(int,(input().split())))

l=[]

for i in a:

    if i not in l:

        l.append(i)

l.sort()

print(\*l)

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

Ques:72(140)

Given 2 numbers N and M followed by N numbers and M numbers, print the common numbers in sorted order.

Input Size : N,M <= 100000 (ie do it in O(n) time complexity)

Sample Testcase :

INPUT

5 4

1 2 3 4 5 1 2 3 4

OUTPUT

1 2 3 4

Test Cases:

Input 1:

1 1

2 2

Output 1:

2

Input 2:

4 4

2 3 4 5 6 5 4 3

Output 2:

3 4 5

Input 3:

2 4

9 8 2 3 9 8

Output 3:

8 9

Input 4:

3 2

1 2 6 6 2

Output 4:

2 6

Input 5:

3 3

1 2 3 4 3 1

Output 5:

1 3

Input 6:

2 2

1 2 1 2

Output 6:

1 2

Input 7:

3 4

1 2 3 4 2 3

Output 7:

2 3

Input 8:

2 3

4 5 6 7 5

Output 8:

5

Input 9:

3 4

3 4 5 5 6 7

Output 9:

5

Input 10:

5 4

4 5 6 5 8 7 7 8 9

Output 10:

5 7 8

Solution:

n,k=map(int,(input().split()))

a=list(map(int,(input().split())))

l=[]

for i in a:

    if(a.count(i)>1):

        if i not in l:

            l.append(i)

l.sort()

print(\*l)

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

Ques:73(128)

Given a binary number convert it into octal format.

Sample Testcase :

INPUT

1100100

OUTPUT

144

Test Cases:

Input 1:

11111

Output 1:

37

Input 2:

11001

Output 2:

31

Input 3:

111000

Output 3:

70

Input 4:

110011

Output 4:

63

Input 5:

001100

Output 5:

14

Input 6:

110111

Output 6:

67

Input 7:

110110

Output 7:

66

Input 8:

1000001

Output 8:

101

Input 9:

101010

Output 9:

52

Input 10:

111111

Output 10:

77

Solution:

s=input()

b=int(s,2)

o=oct(b)

print(o[2:])

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

Ques:74(163)

Given an array of numbers and another number k. Find whether K exists and the number of time k repeats. If it does not exist just print no.

Input Size : |N| <= 1000000

Sample Testcase :

INPUT

5 3

3 3 4 4 7

OUTPUT

yes 2

Test Cases:

Input 1:

5 4

3 4 4 4 1

Output 1:

yes 3

Input 2:

4 3

2 3 4 3

Output 2:

yes 2

Input 3:

4 2

2 2 2 2

Output 3:

yes 4

Input 4:

6 9

8 9 3 3 2 3

Output 4:

yes 9

Input 5:

5 8

8 8 8 8 8

Output 5:

yes 5

Input 6:

4 12

1 12 3 12

Output 6:

yes 2

Input 7:

3 3

1 2 4

Output 7:

no

Input 8:

4 5

1 2 3 3

Output 8:

no

Input 9:

2 2

12 11

Output 9:

no

Input 10:

4 3

2 3 4 5

Output 10:

yes 1

Solution:

n,k=map(int,(input().split()))

a=list(map(int,(input().split())))

for i in a:

    if(i==k):

        print('yes',end=" ")

        print(a.count(i))

        break

else:

    print('no')

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

Ques:75(33)

Given a number N followed by N elements, find the second smallest element.If it cannot be found then print -1

Input Size : N <= 100000 (ie do it in O(log n) time complexity)

Sample Testcase :

INPUT

5

1 2 3 4 5

OUTPUT

2

Test Cases:

Input 1:

8 7 6 0

Output 1:

6

Input 2:

4

6 9 67 54

Output 2:

9

Input 3:

3

-1 -2 -3

Output 3:

-2

Input 4:

4

9 8 -2 -1

Output 4:

-2

Input 5:

5

6 67 87 9 32

Output 5:

9

Input 6:

4

1 4 5 6

Output 6:

4

Input 7:

4

1 89 76 45

Output 7:

45

Input 8:

3

9 8 6

Output 8:

8

Input 9:

2

1 7

Output 9:

7

Input 10:

4

2 3 5 6

Output 10:

3

Solution:

n=int(input())

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

l.sort()

if(l[0]==l[1]):

    print('-1')

else:

    print(l[1])

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

**Ques:76(169)**

Given 2 numbers N,K and an array of N strings, find if any K consecutive strings are same.

Input Size : K <= N <= 1000

Sample Testcase :

INPUT

5 3

code

overload

vishal

vishal

vishal

OUTPUT

yes

Test Cases:

Input 1 :

12 1

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

HeuTNeVayJ

Output 1 :

yes

Input 2 :

21 2

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

DgwBJiYAlJ

Output 2 :

yes

Input 3 :

19 13

gFpPajrjmA

oDFrvlsHcZ

oDFrvlsHcZ

yCloTApAdp

oDFrvlsHcZ

oYpmzahkoq

oDFrvlsHcZ

rETdQUNrjx

uuSzoIeTyC

mgUTKODsvs

CETgjFJhVt

KCaZHxAYYa

oDFrvlsHcZ

oDFrvlsHcZ

NprlxmLJkK

oDFrvlsHcZ

JBLllBpgBW

AzMMSzHrlp

oDFrvlsHcZ

Output 3 :

no

Input 4 :

26 1

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

EUzxYzymKw

Output 4 :

yes

Input 5 :

28 22

fdIfYsPSgd

WKOiqjeiuW

PkjSlJKiRW

ECFGwQuhmy

AuTXcarqdW

kFuPAAQGbO

PkjSlJKiRW

pePprakOqx

BArDZvBfNq

pdlsmvJJUs

PkjSlJKiRW

xjUUZekrUp

UjEeINwljt

PkjSlJKiRW

TUKOlibTGD

FeDmeTQPOD

PkjSlJKiRW

PkjSlJKiRW

PkjSlJKiRW

eOHgJBebpX

CKGPdpMPpc

PkjSlJKiRW

PkjSlJKiRW

PkjSlJKiRW

KDkOPFUmCI

PkjSlJKiRW

PkjSlJKiRW

PkjSlJKiRW

Output 5 :

no

Input 6 :

26 6

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

GvIHZRKWBL

Output 6 :

yes

Input 7 :

26 23

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

YnijQaFSkN

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

IdgZQuhqJm

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

yQDhQjKEcu

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

TeYztcmAYF

fAdbKNIdoT

fAdbKNIdoT

fAdbKNIdoT

Output 7 :

no

Input 8 :

21 15

eJsmLOWVxz

KYFTqUlwoI

MYUyvlVExC

uOoCAxQPpB

vcOBNUKIkN

vDWEANmzyT

bRsgptFMgo

vDWEANmzyT

vDWEANmzyT

eTWwVPRzUz

vDWEANmzyT

vDWEANmzyT

vDWEANmzyT

vDWEANmzyT

UYergqfViT

vDWEANmzyT

QQrBMjjmXU

vDWEANmzyT

ccHgcXgBod

vDWEANmzyT

vLMqjLGdiV

Output 8 :

no

Input 9 :

23 21

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

NnnNJLIiWr

jDhKKZXlPU

jDhKKZXlPU

AUhGcAoWnN

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

jDhKKZXlPU

Output 9 :

yes

Input 10 :

16 1

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

vkrKkaYmAb

Output 10 :

yes

Input 11 :

29 13

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

QOKPsoGkUz

vsGBcDmLwf

vsGBcDmLwf

OmuCKZBTgU

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

efQRCbayfF

ejxirGXlHD

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

vsGBcDmLwf

XXFsOyXvHY

Output 11 :

yes

Input 12 :

13 11

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

WjLPhBeiZt

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

rakLpCMIKO

Output 12 :

yes

Solution:

import re

n,k=map(int,(input().split()))

l=[]

for i in range(n):

    s=input()

    l.append(s)

l1=str(l)

a=re.findall(r'\w+',l1)

d=list(map(str,a))

found = False

for i in d:

   if(d.count(i)>=k):

       found = True

       break

if found:

   print("yes")

else:

   print("no")

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

Ques:77(166)

Given a number N, find the number of ones in its binary representation.

Sample Test Case :

INPUT

276

OUTPUT

3

Test Cases:

Input 1:

346

Output 1:

5

Input 2:

45

Output 2:

4

Input 3:

678

Output 3:

5

Input 4:

456

Output 4:

4

Input 5:

203

Output 5:

5

Input 6:

20

Output 6:

2

Input 7:

12

Output 7:

2

Input 8:

99

Output 8:

4

Input 9:

80

Output 9:

2

Input 10:

06

Output 10:

2

Solution:

n=int(input())

b=bin(n)

for i in b:

a=b.count('1')

print(a)

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

Ques:78(129)

Given a binary number convert it to hexadecimal.

Sample Testcase :

INPUT

1100100

OUTPUT

64

Test Cases:

Input 1:

1111

Output 1:

f

Input 2:

1011

Output 2:

b

Input 3:

1100

Output 3:

c

Input 4:

1010

Output 4:

a

Input 5:

110011

Output 5:

33

Input 6:

0011

Output 6:

3

Input 7:

1001

Output 7:

9

Input 8:

1101

Output 8:

d

Input 9:

11111

Output 9:

1f

Input 10:

11110

Output 10:

1e

Solution:

s=input()

b=int(s,2)

h=hex(b)

hexa=h[2:]

print(hexa)

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

Ques:79(189)

A number is given as input.Find the maximum number that can be formed using the digits.

Input Size : N <= 10000000

Sample Testcase :

INPUT

4123

OUTPUT

4321

Test Cases:

Input 1:

1982

Output 1:

9821

Input 2:

2987634

Output 2:

9876432

Input 3:

775532

Output 3:

775532

Input 4:

09123

Output 4:

93210

Input 5:

878787

Output 5:

888777

Input 6:

98989

Output 6:

99988

Input 7:

12345

Output 7:

54321

Input 8:

6789

Output 8:

9876

Input 9:

345677

Output 9:

776543

Input 10:

77869

Output 10:

98776

Solution:

n=input()

l=list(map(int,str(n)))

l.sort(reverse=True)

print(\*l,sep="")

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

Ques:80(164)

Print the position of first 1 from right to left, in binary representation of an Integer.

Sample Testcase :

INPUT

18

OUTPUT

2

Test Cases:

Input 1:

24

Output 1:

4

Input 2:

23

Output 2:

1

Input 3:

2

Output 3:

2

Input 4:

304

Output 4:

5

Input 5:

345

Output 5:

1

Input 6:

45

Output 6:

1

Input 7:

34

Output 7:

2

Input 8:

78

Output 8:

2

Input 9:

08

Output 9:

4

Input 10:

9

Output 10:

1

Solution:

s=int(input())

b=bin(s)

d=b[2:]

r=d[::-1]

print(r.index('1')+1)

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

**Ques:81(122)**

Given an array of N elements switch(swap) the element with the adjacent element and print the output.

Sample Testcase :

INPUT

5

3 2 1 2 3

OUTPUT

2 3 2 1 3

Test Cases:

Input 1 :

29

75 99 31 47 46 69 34 21 87 75 63 12 31 97 24 55 8 33 42 72 72 30 32 39 36 48 31 21 93

Output 1 :

99 75 47 31 69 46 21 34 75 87 12 63 97 31 55 24 33 8 72 42 30 72 39 32 48 36 21 31 93

Input 2 :

12

6 37 74 92 99 60 5 76 20 86 73 86

Output 2 :

37 6 92 74 60 99 76 5 86 20 86 73

Input 3 :

28

98 97 61 61 80 52 97 57 46 78 71 24 71 63 10 48 69 41 94 1 77 3 11 20 69 96 67 89

Output 3 :

97 98 61 61 52 80 57 97 78 46 24 71 63 71 48 10 41 69 1 94 3 77 20 11 96 69 89 67

Input 4 :

23

10 75 92 46 78 17 55 33 23 40 21 60 3 76 18 53 11 73 22 66 53 67 91

Output 4 :

75 10 46 92 17 78 33 55 40 23 60 21 76 3 53 18 73 11 66 22 67 53 91

Input 5 :

10

87 26 13 56 9 48 94 92 19 8

Output 5 :

26 87 56 13 48 9 92 94 8 19

Input 6 :

23

28 21 55 82 58 72 40 61 19 66 88 55 80 23 72 36 63 48 87 72 96 8 3

Output 6 :

21 28 82 55 72 58 61 40 66 19 55 88 23 80 36 72 48 63 72 87 8 96 3

Input 7 :

19

96 53 17 85 34 47 38 61 46 25 85 71 70 41 80 69 28 87 13

Output 7 :

53 96 85 17 47 34 61 38 25 46 71 85 41 70 69 80 87 28 13

Input 8 :

12

63 88 38 27 8 39 43 95 21 39 74 6

Output 8 :

88 63 27 38 39 8 95 43 39 21 6 74

Input 9 :

12

52 70 27 23 99 46 31 45 32 21 45 49

Output 9 :

70 52 23 27 46 99 45 31 21 32 49 45

Input 10 :

26

54 24 26 55 77 38 51 26 37 12 96 18 78 87 59 14 71 93 16 68 52 87 76 72 15 74

Output 10 :

24 54 55 26 38 77 26 51 12 37 18 96 87 78 14 59 93 71 68 16 87 52 72 76 74 15

Input 11 :

21

53 90 94 12 76 98 60 95 25 75 3 36 5 97 73 48 1 94 55 26 76

Output 11 :

90 53 12 94 98 76 95 60 75 25 36 3 97 5 48 73 94 1 26 55 76

Input 12 :

19

47 6 32 61 8 26 39 4 28 27 4 49 21 97 89 42 59 54 86

Output 12 :

6 47 61 32 26 8 4 39 27 28 49 4 97 21 42 89 54 59 86

Solution:

a=int(input())

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

for i in range(0,a-1,2):

    n[i],n[i+1] = n[i+1], n[i]

print(\*n)

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

**Ques:82(200)**

Given 2 arrays print 'yes' if they are mirror images of each other,otherwise 'no'.

Input Size : N <= 1000000

Sample Testcase :

INPUT

4

1 2 3 4

4 3 2 1

OUTPUT

yes

Test Cases:

Input 1 :

21

4 64 25 26 38 1 41 23 21 2 32 32 66 46 66 55 79 56 9 23 73

73 23 9 56 79 55 66 46 66 32 32 2 21 23 41 1 38 26 25 64 4

Output 1 :

yes

Input 2 :

18

26 48 82 48 94 87 39 77 33 90 90 51 75 97 11 88 26 29

29 26 88 11 97 75 51 90 90 33 77 39 87 94 48 82 48 26

Output 2 :

yes

Input 3 :

25

43 85 25 49 85 23 79 28 41 76 24 38 28 22 5 89 10 77 20 22 37 47 4 89 31

31 89 4 47 37 22 20 77 10 89 5 22 28 38 24 76 41 28 79 23 85 49 25 85 43

Output 3 :

yes

Input 4 :

20

48 23 74 75 43 77 20 90 19 44 4 41 91 42 34 84 57 22 14 70

70 14 22 57 84 34 42 91 41 4 44 19 90 20 77 43 75 74 23 48

Output 4 :

yes

Input 5 :

24

7 70 23 38 21 8 65 92 26 83 69 24 21 85 19 92 28 75 15 46 54 71 82 62

62 82 71 54 46 15 75 28 92 19 85 21 24 69 83 26 92 65 8 21 38 23 70 7

Output 5 :

yes

Input 6 :

19

22 85 52 4 25 91 4 17 13 32 5 55 89 82 80 84 22 36 26

26 36 22 84 80 82 89 55 5 32 13 17 4 91 25 4 52 85 22

Output 6 :

yes

Input 7 :

22

98 73 17 42 56 64 30 3 76 89 87 15 93 72 36 14 77 36 88 77 91 66

98 73 17 42 56 64 30 3 76 89 87 15 93 72 36 14 77 36 88 77 91 66

Output 7 :

no

Input 8 :

21

38 33 91 61 99 11 99 38 98 80 73 29 88 72 50 91 82 20 30 65 39

39 65 30 20 82 91 50 72 88 29 73 80 98 38 99 11 99 61 91 33 38

Output 8 :

yes

Input 9 :

20

29 75 5 43 53 86 19 69 39 30 44 83 49 2 89 76 91 50 32 39

39 32 50 91 76 89 2 49 83 44 30 39 69 19 86 53 43 5 75 29

Output 9 :

yes

Input 10 :

12

97 82 65 64 60 30 38 78 46 34 44 29

29 44 34 46 78 38 30 60 64 65 82 97

Output 10 :

yes

Input 11 :

11

54 8 54 83 46 96 51 86 51 28 57

57 28 51 86 51 96 46 83 54 8 54

Output 11 :

yes

Input 12 :

17

78 7 49 62 1 1 40 46 94 97 62 63 60 19 16 81 66

78 7 49 62 1 1 40 46 94 97 62 63 60 19 16 81 66

Output 12 :

no

Solution:

n=int(input())

a=list(map(int,(input().split())))

b=list(map(int,(input().split())))

if(a==b[::-1]):

print('yes')

else:

print('no')

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

Q**ues:83(195)**

Given a string S, print 'yes' if the strings 'GUVI' and 'GEEK' is present case-sensitively in the string else print 'no'.

Input Size : 1 <= 100

Sample Testcase :

INPUT

Vishal\_Sundar prepared this question

OUTPUT

No

Test Cases:

Input 1 :

IWCGUVICXDYAMB

Output 1 :

Yes

Input 2 :

QJHPFHINEDW

Output 2 :

No

Input 3 :

ODKEXJVMWH

Output 3 :

No

Input 4 :

NJHZYEOMIDGEEKSYOBVRTHXV

Output 4 :

Yes

Input 5 :

TDMMHSTJFQJRTPJ

Output 5 :

No

Input 6 :

KNGUVIBDVQYPCINTEHNQJ

Output 6 :

Yes

Input 7 :

LDOFEBLIXWYXBQGTR

Output 7 :

No

Input 8 :

CAKHQTURWCJ

Output 8 :

No

Input 9 :

ILXPWUMWTCYDPVSN

Output 9 :

No

Input 10 :

FIQIAFGUVKXZYM

Output 10 :

No

Input 11 :

PRJYXIRRNGEEKHAFSDD

Output 11 :

Yes

Input 12 :

LKBZINTRTGEEXKBMAUF

Output 12 :

No

Solution:

import re

s=input()

a=re.compile('GUVI|GEEK')

if a.search(s): print("Yes")

else: print("No")

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

**Ques:84(178)**

Given a string and a number K.Print every kth character from the beginning.

Sample Testcase :

INPUT

string 3

OUTPUT

r g

Test Cases:

Input 1 :

PXVNXBXTPARHVRWKZYOS 9

Output 1 :

P Y

Input 2 :

KWIJGMJIGKPOPTVF 10

Output 2 :

K

Input 3 :

WLVPMYMNZTIISMRUUYA 8

Output 3 :

N U

Input 4 :

ASRGAYLXSP 5

Output 4 :

A P

Input 5 :

JJOBPYQYDMNSU 11

Output 5 :

N

Input 6 :

WUOFZHNHASC 9

Output 6 :

A

Input 7 :

DODQBUWPWXWX 6

Output 7 :

U X

Input 8 :

MHTNIUDTYLOJPLYNXQGWGS 20

Output 8 :

W

Input 9 :

VIJAGSIIXFJMTPVAI 5

Output 9 :

G F V

Input 10 :

CPVNTPRUDMIBILERGUUL 10

Output 10 :

M L

Input 11 :

MTWBZTYEVGFTA 10

Output 11 :

G

Input 12 :

LRKAZKQZFSM 2

Output 12 :

R A K Z S

Solution:

s,n=input().split(maxsplit=1)

k=int(n)

l=list(s)

for i in(k,len(l)):

    f=l[(k-1)::k]

print(\*f)

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

**Ques:85(94)**

Given 2 numbers N and K.Print the number of occurrences of K in N.If K is not found print '-1'.

Input Size : 1 <= N <= 100000, 0 <= K <= 9

Sample Testcase :

INPUT

1000 0

OUTPUT

3

Test Cases:

Input 1 :

98622 2

Output 1 :

2

Input 2 :

6455 5

Output 2 :

2

Input 3 :

61750 6

Output 3 :

1

Input 4 :

8895 8

Output 4 :

2

Input 5 :

8847 1

Output 5 :

-1

Input 6 :

38189 1

Output 6 :

1

Input 7 :

81581 5

Output 7 :

1

Input 8 :

48873 7

Output 8 :

1

Input 9 :

58883 8

Output 9 :

3

Input 10 :

38220 8

Output 10 :

1

Input 11 :

21767 3

Output 11 :

-1

Input 12 :

49260 8

Output 12 :

-1

Solution:

n,k=map(int,(input().split()))

l=list(map(int,str(n)))

for i in l:

        if(i==k):

            print(l.count(i))

            break

else:

    print('-1')

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

**Ques:86(179)**

Given a string and a number K, change every kth character to uppercase from beginning in string.

Sample Testcase :

INPUT

string 2

OUTPUT

sTrInG

Test Cases:

Input 1 :

ltbitkouadnhbikotxbot 15

Output 1 :

ltbitKouadnhbiKotxbot

Input 2 :

jliorjnhzzyojaa 4

Output 2 :

jliOrjnHzzyOjaa

Input 3 :

hnovytwdet 9

Output 3 :

hnovytwdEt

Input 4 :

sfdpwokrjqrjuyybfbq 13

Output 4 :

sfdpwokrjqrjUyybfbq

Input 5 :

xbsfunylypom 11

Output 5 :

xbsfunylypOm

Input 6 :

dhowwvpqikkvoduh 5

Output 6 :

dhoWWvpqiKKvodUh

Input 7 :

vjrzyxnvbfcuqioqwbuj 10

Output 7 :

vJrzyxnvbFcuqioqwbuJ

Input 8 :

atawnxakwewomplb 4

Output 8 :

ataWnxaKWeWOmplB

Input 9 :

etwdxanjnlz 3

Output 9 :

etWdxANjNlz

Input 10 :

dudfekiydrfecdck 6

Output 10 :

dudfEKiydrfEcdcK

Input 11 :

dnesoyswfqnkk 12

Output 11 :

dnesoyswfqnKK

Input 12 :

sbwsenwdiwrdwgp 1

Output 12 :

SBWSENWDIWRDWGP

Solution:

s,n=input().split(maxsplit=1)

k=int(n)

l=list(s)

for i in(k,len(l)):

    f=l[(k-1)::k]

a=[]

for i in l:

    if i in f:

        a.append(i)

b=[]

for j in l:

    if j in a:

        d=str(j)

        d1=d.upper()

        b.append(d1)

    if j not in a:

        d2=str(j)

        b.append(d2)

e="".join(b)

print(e)

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

**Ques:88(305)**

Given a number 'N' print the sum of each digit to the power of number of digits in given input.

Example :

Input => 1234

=> ( 1 ^ 4 ) + ( 2 ^ 4 ) + ( 3 ^ 4 ) + ( 4 ^ 4 )

=> 1 + 16 + 81 + 256

Output => 354

N <=100000000000

Sample Testcase :

INPUT

1234

OUTPUT

354

Test Cases:

Input 1 :

34180

Output 1 :

34036

Input 2 :

6521

Output 2 :

1938

Input 3 :

60531

Output 3 :

11145

Input 4 :

28179

Output 4 :

108657

Input 5 :

59425

Output 5 :

66355

Input 6 :

92614

Output 6 :

67882

Input 7 :

11973

Output 7 :

76101

Input 8 :

8261

Output 8 :

5409

Input 9 :

47220

Output 9 :

17895

Input 10 :

48055

Output 10 :

40042

Input 11 :

29675

Output 11 :

86789

Input 12 :

48595

Output 12 :

99091

Solution:

import re

s=input()

l=len(s)

a=re.findall(r'\d',s)

d=list(map(int,a))

l1=[]

for i in d:

    n=i\*\*l

    l1.append(n)

sum=0

for j in l1:

    sum=sum+j

print(sum)

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

**Ques:88(141)**

Given an array print the number of subarrays that can be formed with it.

Input Size : N <= 100000

Sample Testcase :

INPUT

5

1 2 3 2 1

OUTPUT

15

Test Cases:

Input 1 :

6

77612 10134 42213 1855 52431 63978

Output 1 :

21

Input 2 :

4

77922 4569 35755 27315

Output 2 :

10

Input 3 :

6

55108 83715 17133 58266 96482 88039

Output 3 :

21

Input 4 :

3

95213 2682 26427

Output 4 :

6

Input 5 :

6

327 60693 39862 31361 14159 41198

Output 5 :

21

Input 6 :

3

62295 87330 21411

Output 6 :

6

Input 7 :

3

42907 71290 70589

Output 7 :

6

Input 8 :

6

87782 59315 42668 63969 8226 49274

Output 8 :

21

Input 9 :

3

58643 30006 32716

Output 9 :

6

Input 10 :

4

47794 42986 39252 19537

Output 10 :

10

Input 11 :

3

35202 39261 3284

Output 11 :

6

Input 12 :

4

93734 98700 62385 73100

Output 12 :

10

Solution:

n=int(input())

a=list(map(int,(input().split())))

l=[]

for i in range(len(a)+1):

    for j in range(i+1,len(a)+1):

        s=a[i:j]

        l.append(s)

print(len(l))

Better Solution:

n=int(input())

a=list(map(int,(input().split())))

print(n\*(n+1)/ 2)

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

**Ques:89(168)**

Given a number N and an array of N strings, find if two consecutive words are same.

Input Size : N <= 1000

Sample Testcase :

INPUT

5

code

overload

vishal

sundar

anish

OUTPUT

No

Test Cases:

Input 1 :

22

hCdvlLQYLN

oiIgLfXTDw

EWHSDIGyZZ

EWHSDIGyZZ

nHmYSmmamA

EWHSDIGyZZ

OPROvWjCZW

CNVayfNdBm

EWHSDIGyZZ

VGSkjlNqbw

KcnVxrShZG

oBxAvgOfZP

EWHSDIGyZZ

lauvPpehQp

BUGruBaUra

FBdnkTfLWO

EWHSDIGyZZ

EWHSDIGyZZ

qlIaMLEDNA

EWHSDIGyZZ

tBNMpkTLoU

vqCBmfgEkD

Output 1 :

yes

Input 2 :

16

geWESbwDIH

BsEVxewani

BsEVxewani

rUovCXzzwY

BsEVxewani

tOuBETGUnL

MxuCCCPQoa

AYHkxhIufN

YWjoDWeZEE

BsEVxewani

truGiqMmXI

EUpDaLtEcW

BsEVxewani

BsEVxewani

BsEVxewani

CZIAAgPGzm

Output 2 :

yes

Input 3 :

12

MBLGoYkvoO

XjTXCvYAWb

pcsXlqPNqN

usUMeYFKcb

nUCBqcUGfg

PeuLLjwIhe

pcsXlqPNqN

XcPKrxFplG

SgxIAlxWjZ

pcsXlqPNqN

uppBvvJMdg

iYZnAgJLlD

Output 3 :

yes

Input 4 :

23

xtuaQQHjQy

wwOGvyGBqz

ewXubLXyHf

wwOGvyGBqz

EGifvKrinp

MZCGnpcujo

FZoAsYSmjk

wwOGvyGBqz

wwOGvyGBqz

wwOGvyGBqz

wwOGvyGBqz

ngcvKmmSFs

IpbTRceDLR

wwOGvyGBqz

wwOGvyGBqz

tovzYHlAno

WDhrxmOplz

DmSMmZWVHH

VqFQdVVSAz

yogGVmvZrJ

ZXhTacIiXy

GlaaRtkIyW

wwOGvyGBqz

Output 4 :

yes

Input 5 :

15

rHApBNgTTN

DNucrPtTEC

aOsrDOMQdp

vruqZuaTWd

qEVAscCvhq

rHApBNgTTN

rHApBNgTTN

rHApBNgTTN

zQAlkuanpv

bMGdrZIEVO

HpkeujfDvj

kOzxqhVChH

rHApBNgTTN

XIGMrbTVpn

pJNAOEigyS

Output 5 :

yes

Input 6 :

23

qlxiCWQuhr

VcSQCVODze

yiNPxhqcfk

NCNXUBbnvW

ciOIMVRSQz

VcSQCVODze

VcSQCVODze

OFbnQeNuSa

uWGcBnltry

VcSQCVODze

rVrkEPLQwA

VcSQCVODze

VcSQCVODze

DqFEIeqjmh

fIuWiXlgsl

rCRtMBkCLI

csfEIDwqsj

VcSQCVODze

VcSQCVODze

VcSQCVODze

VcSQCVODze

VcSQCVODze

VcSQCVODze

Output 6 :

yes

Input 7 :

24

LKlVjEDbbb

kzJuOKnvzS

FTMwLblPld

OtvdbgOoZy

gfLMSWFXgL

axdwjjWSZJ

LKlVjEDbbb

LKlVjEDbbb

OAWjLGFpkF

grBpKsPHgp

LKlVjEDbbb

YmtGziRqQK

LKlVjEDbbb

LKlVjEDbbb

mGOIDbrchm

LKlVjEDbbb

KAbnqvlUJl

uSlngCXLpu

LKlVjEDbbb

JOjxIhnkJO

Uhmubhsnen

LKlVjEDbbb

jCofifxZkX

zLLHDYovWl

Output 7 :

yes

Input 8 :

26

WoRkcvzjZG

yFqoNWFSPd

TlEUFFdIfN

yFqoNWFSPd

yFqoNWFSPd

yFqoNWFSPd

yFqoNWFSPd

QCQhxIVxdD

yFqoNWFSPd

yFqoNWFSPd

yFqoNWFSPd

yFqoNWFSPd

fUtHxwjLas

yFqoNWFSPd

inRfVFSEyA

ZTuFEyLqAx

SbPdyvKiqQ

yFqoNWFSPd

bKgRljnpQH

pJOkNLNPlz

yFqoNWFSPd

yFqoNWFSPd

dcgUodAhaS

yFqoNWFSPd

yFqoNWFSPd

cXsVakaUHG

Output 8 :

yes

Input 9 :

16

zYYCtejPRZ

ltbOnzFEYI

dxsPAPlgGw

ZGntscmFWF

ZGntscmFWF

ZGntscmFWF

aqPFPCyEYf

ZGntscmFWF

FxnrllbRIa

hORcBzzRTZ

XqJGBTqbcS

EaByGivIkS

ZGntscmFWF

UoAYCaGBvm

ZGntscmFWF

AJGqJTZRsX

Output 9 :

yes

Input 10 :

10

FIESMIroMu

NOJcZccqKe

SzjuvNEIbJ

hutmynkTdJ

ZgPqAeGTMY

zdtAOnvFUt

tbBmEDpMpU

PgmCglaWae

rYgGfRiFLY

TJFYjCCJDu

Output 10 :

no

Input 11 :

14

ZSjCoMSZuX

GSOPxCpzFi

jhDFAjxttN

jFddZnZJjp

ZSjCoMSZuX

SfnuDuBnPh

ZSjCoMSZuX

WREkdsWCQY

ZSjCoMSZuX

yzfvIbFRJQ

AmnpiHBtRx

ZSjCoMSZuX

hQYkeygEze

imxNPqGBuP

Output 11 :

yes

Input 12 :

16

ZdVSvUWdpA

zInTszXtoZ

nsUnnunKFY

ZdVSvUWdpA

IitmiOWwqy

VkaxCpElbd

UFluCICamk

ZdVSvUWdpA

zCtjfaRCJI

ZdVSvUWdpA

qvAJNyPqrd

PmsIFDuKXw

eKtrPRrUmX

ZdVSvUWdpA

PXWVpRzwPT

ZdVSvUWdpA

Output 12 :

yes

Solution:

import re

n=int(input())

l=[]

for i in range(n):

    s=input()

    l.append(s)

l1=str(l)

a=re.findall(r'\w+',l1)

d=list(map(str,a))

for i in d:

    if(d.count(i)==2):

        print('yes')

        break

else:

print('no')

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

**Ques:90(390)**

Vasanth is working at GUVI. He has been taking leave often in the past couple of weeks and his manager, who also happens to be his friend, is worried that Vasanth might be exceeding his number of paid holidays, which might be a black mark in Vasanth monthly report. The manager analysis Vasanth’s attendance register and decides to warn him beforehand. The attendance register has a ‘P’ marked for present and an ‘A’ marked for absent. Vasanth will be blacklisted if his attendance falls below 25%. Your task is to help the manager find out whether Vasanth could be blacklisted or not.

Input Description:

First line contains the number of entries in the attendance register, followed by space separated values of attendance (‘P’ or ‘A’)

Output Description:

‘Blacklisted’ if attendance is below 25% ‘Not Blacklisted’ if attendance is above 25%

Sample Input :

5

P P A A A

Sample Output :

Not Blacklisted

Test Cases:

Input 1 :

25

P A A A A A A A P P A A P A A A P A P A P A A P A

Output 1 :

Not Blacklisted

Input 2 :

10

A A A A A P A A A A

Output 2 :

Blacklisted

Input 3 :

21

P A A A A A A A A A A A P P A A A A A A A

Output 3 :

Blacklisted

Input 4 :

19

A A A A A A A A A A P P A A A A P A A

Output 4 :

Blacklisted

Input 5 :

15

A A P P P A A A A P A A A A A

Output 5 :

Not Blacklisted

Input 6 :

11

A A A A P A A P P A A

Output 6 :

Not Blacklisted

Input 7 :

15

A A A A A A A A A A A A P A A

Output 7 :

Blacklisted

Input 8 :

28

A P A P A P P P A A A A A A A A A A A A A A A P A A A A

Output 8 :

Blacklisted

Input 9 :

22

A A A A A A A A A A A A P P P A P A P A A A

Output 9 :

Blacklisted

Input 10 :

26

A A A A P A A A A A A P A A A A A A A A A A A P A A

Output 10 :

Blacklisted

Input 11 :

21

P A A A A A P A A P A A P P A A A P A A A

Output 11 :

Not Blacklisted

Input 12 :

16

A A A P A P A A A A P A A A A A

Output 12 :

Blacklisted

Solution:

n=int(input())

l=list(map(str,input().split()))

t=[]

for i in l:

    if(i=='P'):

        t.append(100)

    if(i=='A'):

        t.append(0)

a=sum(t)//len(l)

if(a<=25):

    print('Blacklisted')

else:

    print('Not Blacklisted')

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

**Ques:91(674)**

You are given with string of words,we have to arrange them in reverse saturated order.

Input Description:

You are given a string ‘s’.

Output Description:

Print the reverse saturated order

Sample Input :

I am kohli fan

Sample Output :

I ma ilhok naf

Test Cases:

Input 1 :

sNa fRTkhPrPl ORSXCFV Lqgefu tOedQB f

Output 1 :

aNs lPrPhkTRf VFCXSRO ufegqL BQdeOt f

Input 2 :

OEMWxrH eieHWUBC mUiZbeSK getJzag HxvZjvZXu zGwBL lVvctIvhC qyuSFTp MwAGO

Output 2 :

HrxWMEO CBUWHeie KSebZiUm gazJteg uXZvjZvxH LBwGz ChvItcvVl pTFSuyq OGAwM

Input 3 :

A LRkMUg vNq qQnYvU LepM gJxztTRJ Ss SrqxNrtJ

Output 3 :

A gUMkRL qNv UvYnQq MpeL JRTtzxJg sS JtrNxqrS

Input 4 :

y Tlpq nowARZxJy hziGP szWYSU jCsAXnQD VVZdFoLl

Output 4 :

y qplT yJxZRAwon PGizh USYWzs DQnXAsCj lLoFdZVV

Input 5 :

TLsq WpDXeoFB QBrjg JwZJHrCN c Ouytb acK YowN

Output 5 :

qsLT BFoeXDpW gjrBQ NCrHJZwJ c btyuO Kca NwoY

Input 6 :

iiuuUTUdO Vp iIx AQesx Wl JS

Output 6 :

OdUTUuuii pV xIi xseQA lW SJ

Input 7 :

fhPQTzhID cbFfRjsR TAS xxUEaZtkf iVMxnn skkE pcMmHShsS

Output 7 :

DIhzTQPhf RsjRfFbc SAT fktZaEUxx nnxMVi Ekks SshSHmMcp

Input 8 :

EvQCaGxzR Gbevioan Ht GvNlOUh cC

Output 8 :

RzxGaCQvE naoivebG tH hUOlNvG Cc

Input 9 :

hVytYjzEU HVInKeVw usSWV dIu XU hpT LtDCeJyX

Output 9 :

UEzjYtyVh wVeKnIVH VWSsu uId UX Tph XyJeCDtL

Input 10 :

BDVm zyLTJltIW lGH szGKULce J yUj WxthSR LfXAvArjB KvvBkT

Output 10 :

mVDB WItlJTLyz HGl ecLUKGzs J jUy RShtxW BjrAvAXfL TkBvvK

Input 11 :

cx kBUmZEk J iJPLPb zQRWJ ChZz UDkU yuWz

Output 11 :

xc kEZmUBk J bPLPJi JWRQz zZhC UkDU zWuy

Input 12 :

vb sD lqNAwz q tiem

Output 12 :

bv Ds zwANql q meit

Solution:

s=input()

a=s.split()

l=[]

for i in a:

    b=i[::-1]

    l.append(b)

print(\*l)

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

**Ques:92(206)**

Given a number N and array of N integers, print the difference between the indices of smallest and largest number(if there are multiple occurances, consider the first occurance).

Input Size : |N| <= 1000000

Sample Testcase :

INPUT

5

3 5 4 4 7

OUTPUT

4

Test Cases:

Input 1 :

5

24 18 18 16 16

Output 1 :

-3

Input 2 :

5

13 16 9 12 16

Output 2 :

-1

Input 3 :

6

12 4 5 7 13 10

Output 3 :

3

Input 4 :

6

20 11 19 11 8 19

Output 4 :

-4

Input 5 :

6

1 11 11 22 7 19

Output 5 :

3

Input 6 :

8

4 21 24 19 6 17 19 11

Output 6 :

2

Input 7 :

5

15 18 13 5 10

Output 7 :

-2

Input 8 :

7

14 22 21 7 23 14 14

Output 8 :

1

Input 9 :

6

14 10 1 19 3 6

Output 9 :

1

Input 10 :

8

17 17 22 22 22 3 24 1

Output 10 :

-1

Input 11 :

8

7 19 11 22 12 17 7 12

Output 11 :

3

Input 12 :

7

4 13 6 19 20 5 6

Output 12 :

4

Solution:

n=int(input())

s=list(map(int,(input().split())))

d=min(s)

d1=max(s)

c=s.index(d)

c1=s.index(d1)

a=c1-c

print(a)

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

**Ques:93(75)**

Given a string S,count the maximum number of times a character repeated in the string.If no character is repeated print '0'.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codekata

OUTPUT

2

Test Cases:

Input 1 :

STQIgFlz

Output 1 :

0

Input 2 :

MqFTVPFTcAClQv

Output 2 :

2

Input 3 :

RpmstqkZIrHv

Output 3 :

0

Input 4 :

XxNegfYbub

Output 4 :

2

Input 5 :

IbyIEni

Output 5 :

2

Input 6 :

JfqMoyciTtW

Output 6 :

0

Input 7 :

CPVtfIb

Output 7 :

0

Input 8 :

kYyexGctkNjMuP

Output 8 :

2

Input 9 :

OgTGNRmAFU

Output 9 :

0

Input 10 :

nZtJlgZvs

Output 10 :

2

Input 11 :

HbGUm

Output 11 :

0

Input 12 :

lfxWczcwibx

Output 12 :

2

Solution:

s=input()

l=[]

for i in s:

    a=s.count(i)

    l.append(a)

d=max(l)

if(d==1):

    print('0')

else:

    print(d)

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

**Ques:94(79)**

Given a number N, print their prime factors in sorted order.

Input Size : 2 <= N <= 100000

Sample Testcase :

INPUT

18

OUTPUT

2 3

Test Cases:

Input 1 :

984

Output 1 :

2 3 41

Input 2 :

2631

Output 2 :

3 877

Input 3 :

3892

Output 3 :

2 7 139

Input 4 :

2938

Output 4 :

2 13 113

Input 5 :

528

Output 5 :

2 3 11

Input 6 :

1049

Output 6 :

1049

Input 7 :

1775

Output 7 :

5 71

Input 8 :

9462

Output 8 :

2 3 19 83

Input 9 :

4627

Output 9 :

7 661

Input 10 :

417

Output 10 :

3 139

Input 11 :

4557

Output 11 :

3 7 31

Input 12 :

2599

Output 12 :

23 113

Solution:

n=int(input())

f=[]

for i in range(1,n+1):

    if(n%i==0):

        f.append(i)

pf=[]

for j in f:

    if j>1:

        for k in range(2,j):

            if(j%k==0):

break

        else:

            pf.append(j)

print(\*pf)

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

**Ques:95(39)**

Given a range (i.e) two numbers L and R count the number of perfect squares within the range (inclusive of L and R).If no perfect square exists within the range print '-1'.

Input Size : L <= R <= 100000(complexity O(n))

Sample Testcase :

INPUT

2 10

OUTPUT

2

Test Cases:

Input 1 :

5791 16738

Output 1 :

53

Input 2 :

7458 48899

Output 2 :

135

Input 3 :

7041 68640

Output 3 :

178

Input 4 :

3873 70959

Output 4 :

204

Input 5 :

8723 46827

Output 5 :

123

Input 6 :

9440 33883

Output 6 :

87

Input 7 :

936 3947

Output 7 :

32

Input 8 :

2510 45947

Output 8 :

164

Input 9 :

573 78207

Output 9 :

256

Input 10 :

8994 29417

Output 10 :

77

Input 11 :

8079 90488

Output 11 :

211

Input 12 :

4109 75130

Output 12 :

210

Solution:

import math

a,b=map(int,(input().split()))

c=(math.floor(math.sqrt(b)) - math.ceil(math.sqrt(a)) + 1)

r=int(c)

if(r==0):

    print(-1)

else:

    print(r)

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

**Ques:96(91)**

Given a string S consisting of only '(' and ')', print 'yes' if it is balanced otherwise print 'no'.

Sample Testcase :

INPUT

(())

OUTPUT

Yes

Test Cases:

Input 1 :

))))(

Output 1 :

no

Input 2 :

)(((()())()((()

Output 2 :

no

Input 3 :

())))

Output 3 :

no

Input 4 :

(((()))))(())(((

Output 4 :

no

Input 5 :

())((

Output 5 :

no

Input 6 :

)(())(

Output 6 :

no

Input 7 :

)))(()(()))(()(

Output 7 :

no

Input 8 :

()(())

Output 8 :

yes

Input 9 :

)())(()()))

Output 9 :

no

Input 10 :

(()())))((((

Output 10 :

no

Input 11 :

(()((())()()(()()))

Output 11 :

no

Input 12 :

(()(((()((

Output 12 :

no

Solution:

s=input()

count=0

c=0

for i in s:

    if(i=='('):

        count=count+1

        continue

for j in s:

    if(j==')'):

        c=c+1

if(count==c):

print('yes')

else:

print('no')

def isValid(s):

   d = {')':'('}

   stack = []

   for i in s:

       if i in d.values():

           stack.append(i)

       elif not stack or stack.pop() != d[i]:

           return False

   return not stack

expr = input()

if (isValid(expr)) :

  print("yes")

else :

  print("no")

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

**Ques:97(47)**

Given 2 numbers N,K and an array of N elements, print the number(s) that has been repeated K times.Print them in ascending order if there are more than one number to be printed.If no element satisfies the pattern then print -1

Input Size : N,K <= 100000

Sample Testcase :

INPUT

5 2

1 2 4 1 2

OUTPUT

1 2

Test Cases:

Input 1 :

13 3

335 335 958 958 623 762 136 762 136 335 335 958 136

Output 1 :

136 958

Input 2 :

11 3

862 862 862 751 751 646 862 500 862 952 500

Output 2 :

-1

Input 3 :

17 3

668 755 605 755 837 755 837 173 755 173 755 173 837 755 755 837 755

Output 3 :

173

Input 4 :

17 3

674 107 674 606 674 310 674 107 107 107 674 577 107 674 310 674 107

Output 4 :

-1

Input 5 :

11 4

89 638 531 483 483 483 531 531 638 716 483

Output 5 :

483

Input 6 :

10 3

668 477 668 726 477 799 477 120 477 668

Output 6 :

668

Input 7 :

14 3

507 743 365 319 365 507 507 365 743 365 660 660 365 365

Output 7 :

507

Input 8 :

14 3

532 63 641 154 532 600 532 154 154 641 532 600 532 641

Output 8 :

154 641

Input 9 :

18 4

399 399 432 399 210 432 835 698 698 399 399 399 432 210 210 698 698 835

Output 9 :

698

Input 10 :

18 4

185 580 608 185 997 563 997 608 580 580 997 185 997 185 580 185 185 185

Output 10 :

580 997

Input 11 :

16 3

658 372 543 4 658 4 372 372 372 580 543 580 580 4 543 372

Output 11 :

4 543 580

Input 12 :

11 4

406 406 506 506 555 506 506 506 860 274 406

Output 12 :

-1

Solution:

n,k=map(int,(input().split()))

a=list(map(int,(input().split())))

l=[]

for i in a:

    if(a.count(i)==k):

        if i not in l:

            l.append(i)

if(len(l)==0):

    print(-1)

else:

l.sort()

    print(\*l)

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

**Ques:98(115)**

Given a number N and an array of N elements, find the Bitwise XOR of the array elements.

Input Size : N <= 100000

Sample Testcase :

INPUT

2

2 4

OUTPUT

6

Test Cases:

Input 1 :

12

872 872 778 872 546 440 766 546 872 872 872 872

Output 1 :

804

Input 2 :

10

772 292 394 292 992 394 328 292 772 772

Output 2 :

136

Input 3 :

12

846 961 242 846 518 518 221 846 846 846 518 961

Output 3 :

359

Input 4 :

14

235 564 564 564 460 200 235 235 200 235 235 200 601 564

Output 4 :

950

Input 5 :

12

19 172 428 428 172 663 428 150 663 150 428 663

Output 5 :

644

Input 6 :

14

543 932 450 543 932 542 450 543 542 932 543 959 543 542

Output 6 :

26

Input 7 :

13

322 101 322 913 913 101 718 322 718 718 913 707 322

Output 7 :

924

Input 8 :

16

185 185 952 952 185 612 952 528 816 952 952 185 816 952 185 952

Output 8 :

885

Input 9 :

11

706 253 337 253 190 337 190 337 581 581 337

Output 9 :

706

Input 10 :

17

52 804 804 52 804 52 353 804 353 16 812 52 353 804 52 804 52

Output 10 :

605

Input 11 :

10

752 118 118 752 911 911 752 911 174 822

Output 11 :

743

Input 12 :

15

763 763 409 409 763 792 763 792 678 409 678 792 314 678 409

Output 12 :

132

Solution:

n=int(input())

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

k=l[0]

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

k=k^(l[i])

print(k)

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

**Ques:99(204)**

Given a number N and array of N integers, find the number which occurs the least number of times.

Input Size : |N| <= 1000000

Sample Testcase :

INPUT

5

3 3 4 4 7

OUTPUT

7

Test Cases:

Input 1 :

19

435 481 459 435 481 916 412 412 667 435 667 459 459 435 156 435 435 916 459

Output 1 :

156

Input 2 :

13

905 905 905 313 162 430 880 880 2 905 905 905 880

Output 2 :

2

Input 3 :

18

38 88 24 801 151 38 30 24 29 24 30 38 371 30 801 88 88 24

Output 3 :

371

Input 4 :

17

977 381 977 819 381 101 101 819 101 948 749 819 101 381 948 101 749

Output 4 :

749

Input 5 :

16

682 261 53 550 203 550 700 700 203 198 700 196 203 53 198 323

Output 5 :

323

Input 6 :

17

192 227 452 452 1 452 192 192 452 693 192 693 452 227 227 1 192

Output 6 :

693

Input 7 :

15

936 936 936 796 16 936 860 802 860 860 860 936 16 936 16

Output 7 :

802

Input 8 :

15

115 929 793 960 929 929 572 572 793 410 725 960 572 725 960

Output 8 :

410

Input 9 :

17

563 909 562 814 814 713 562 563 563 562 563 30 563 563 909 629 562

Output 9 :

629

Input 10 :

15

970 105 265 265 238 871 238 970 871 871 105 871 105 970 871

Output 10 :

238

Input 11 :

15

407 728 263 728 256 539 263 539 668 539 539 407 246 668 539

Output 11 :

246

Input 12 :

13

562 319 562 239 142 318 239 923 395 339 556 923 142

Output 12 :

556

Solution:

n=int(input())

a=list(map(int,(input().split())))

for i in a:

    if(a.count(i)==1):

        print(i)

import collections

n=int(input())

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

freq = collections.Counter(l)

print(freq.most\_common()[-1][0] )

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

**Ques:100(126)**

Given a range[L,R], print the sum of all the odd numbers within the range(inclusive of L and R).

Sample Testcase:

INPUT

5 10

OUTPUT

21

Test Cases:

Input 1 :

61153 714976

Output 1 :

126862778368

Input 2 :

26063 280892

Output 2 :

19555271955

Input 3 :

42070 852592

Output 3 :

181285808391

Input 4 :

62839 935722

Output 4 :

217906761760

Input 5 :

95321 262859

Output 5 :

15002369300

Input 6 :

44244 320036

Output 6 :

25116377440

Input 7 :

38686 107109

Output 7 :

2493986376

Input 8 :

32600 107821

Output 8 :

2640705921

Input 9 :

72105 898753

Output 9 :

200639941425

Input 10 :

70373 206933

Output 10 :

9467365493

Input 11 :

97455 482534

Output 11 :

55835444760

Input 12 :

93269 642833

Output 12 :

101134157933

Solution:

l,u=map(int,(input().split()))

sum=0

for i in range(l,u+1):

    if(i%2!=0):

        sum=sum+i

print(sum)

**Better Approach:**

def sumOdd(n):

   terms = (n + 1)//2

   sum1 = terms \* terms

   return sum1

def suminRange(l, r):

   return sumOdd(r) - sumOdd(l - 1)

l,u=map(int,(input().split()))

print(suminRange(l,u))

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

**Ques:101(46)**

Given a number N and an array of N elements, print all elements lesser than N in descending order.If no element found print -1

Input Size : 1 <= N <= 10000

Sample Testcase :

INPUT

5

2 14 15 14 3

OUTPUT

3 2

Test Cases:

Input 1 :

16

24 20 3 14 4 9 19 24 5 20 2 20 2 20 19 23

Output 1 :

14 9 5 4 3 2 2

Input 2 :

16

22 16 2 21 6 11 20 16 4 24 16 23 18 19 20 11

Output 2 :

11 11 6 4 2

Input 3 :

15

4 5 13 7 9 17 1 12 14 10 2 22 9 9 21

Output 3 :

14 13 12 10 9 9 9 7 5 4 2 1

Input 4 :

11

8 23 10 8 3 6 20 21 6 12 15

Output 4 :

10 8 8 6 6 3

Input 5 :

13

24 5 5 10 8 7 21 21 10 13 18 22 11

Output 5 :

11 10 10 8 7 5 5

Input 6 :

17

9 3 12 13 24 15 17 3 8 10 3 11 1 19 11 14 19

Output 6 :

15 14 13 12 11 11 10 9 8 3 3 3 1

Input 7 :

13

2 18 21 24 18 12 9 19 18 10 1 14 6

Output 7 :

12 10 9 6 2 1

Input 8 :

14

4 19 16 1 10 3 20 22 20 13 12 3 2 24

Output 8 :

13 12 10 4 3 3 2 1

Input 9 :

15

8 13 11 9 16 1 17 13 7 8 10 24 17 21 11

Output 9 :

13 13 11 11 10 9 8 8 7 1

Input 10 :

18

10 19 5 3 6 8 15 17 15 5 7 2 12 4 8 21 6 1

Output 10 :

17 15 15 12 10 8 8 7 6 6 5 5 4 3 2 1

Input 11 :

15

10 10 19 20 22 17 4 24 7 2 7 19 20 22 4

Output 11 :

10 10 7 7 4 4 2

Input 12 :

15

13 17 3 24 3 10 19 21 6 8 23 14 11 11 6

Output 12 :

14 13 11 11 10 8 6 6 3 3

n=int(input())

a=list(map(int,(input().split())))

l=[]

for i in a:

    if(i<n):

        l.append(i)

if(len(l)==0):

    print(-1)

else:

    l.sort(reverse=True)

    print(\*l)

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

Ques:102(90)

Given 2 strings and a number K, check whether they differ exactly by K characters.

Input Size : |s| <= 100000(complexity O(nlogn) or O(n))

Sample Testcase :

INPUT

codekata codeguvi 4

OUTPUT

Yes

Test Cases:

Input 1 :

LympIaXQTg CILQTXaeeggmpy 2

Output 1 :

No

Input 2 :

QxYCdEyMApnyQpHFe ABCEFHMQQYdelnpptxyy 2

Output 2 :

No

Input 3 :

ieWhKSyqLzQoGxqRE EGKLQRSWehilmmoqqxyz 3

Output 3 :

Yes

Input 4 :

JAEfezFEeCyLxWEiB ABCEEEFJLPWeefixyz 3

Output 4 :

No

Input 5 :

HZNUABJaumVVrSprWyt ABHJNSUVVWXZamnprrtuyy 2

Output 5 :

Yes

Input 6 :

siXYYtkAPzVwnmypz AJOPVXYYZikmnpstwyzz 3

Output 6 :

Yes

Input 7 :

QmRrTBJvkYT BHJQRTTYkmrv 4

Output 7 :

No

Input 8 :

WiCczlyRJJBZAu ABCGJJOQRWZZciluyz 4

Output 8 :

No

Input 9 :

pcZrxhBzizJWJlLfkA ABHJJLUWZcfhikklpqrxzz 3

Output 9 :

Yes

Input 10 :

IRmFRwjqJNMeP FIJMNNPRRXejmqw 4

Output 10 :

No

Input 11 :

UOCslaQQjowQ CMOQQQUajlosuwx 3

Output 11 :

Yes

Input 12 :

JXJlQutztDGdFboM DFGJJMQXXbdglottuz 2

Output 12 :

No

Solution:

s1,s2,k=[str(s1) for s1 in input().split()]

l1=list(s1)

l2=list(s2)

K=int(k)

l3=[]

for i in l1:

    if i not in l2:

        l3.append(i)

if(len(l3)==K):

    print('Yes')

else:

    print('No')

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

**Ques:103(138)**

Given a number N and an array of N elements, print the suffix sum of the array.

Input Size : N <= 100000

Sample Testcase :

INPUT

4

2 4 4 2

OUTPUT

12 10 6 2

Test Cases:

Input 1 :

16

24 13 9 9 6 22 16 15 1 6 23 4 8 4 23 8

Output 1 :

191 167 154 145 136 130 108 92 77 76 70 47 43 35 31 8

Input 2 :

18

12 16 6 5 24 2 15 20 13 12 9 24 20 7 6 3 23 17

Output 2 :

234 222 206 200 195 171 169 154 134 121 109 100 76 56 49 43 40 17

Input 3 :

13

8 11 5 18 12 1 23 2 10 8 19 10 24

Output 3 :

151 143 132 127 109 97 96 73 71 61 53 34 24

Input 4 :

14

10 6 12 3 20 11 11 21 13 23 3 16 13 23

Output 4 :

185 175 169 157 154 134 123 112 91 78 55 52 36 23

Input 5 :

17

5 8 23 23 9 21 19 24 14 7 8 6 22 17 13 13 8

Output 5 :

240 235 227 204 181 172 151 132 108 94 87 79 73 51 34 21 8

Input 6 :

13

14 19 24 13 1 10 8 1 18 18 8 1 11

Output 6 :

146 132 113 89 76 75 65 57 56 38 20 12 11

Input 7 :

17

17 3 3 17 20 20 17 6 15 14 12 14 17 22 8 1 17

Output 7 :

223 206 203 200 183 163 143 126 120 105 91 79 65 48 26 18 17

Input 8 :

14

1 12 13 1 6 20 15 2 18 6 12 4 7 22

Output 8 :

139 138 126 113 112 106 86 71 69 51 45 33 29 22

Input 9 :

13

6 5 15 8 18 22 22 23 19 20 21 6 10

Output 9 :

195 189 184 169 161 143 121 99 76 57 37 16 10

Input 10 :

10

10 15 8 6 1 16 21 14 17 21

Output 10 :

129 119 104 96 90 89 73 52 38 21

Input 11 :

18

2 3 23 24 1 4 17 1 9 11 17 2 3 21 14 23 20 14

Output 11 :

209 207 204 181 157 156 152 135 134 125 114 97 95 92 71 57 34 14

Input 12 :

15

23 23 15 11 7 11 4 15 11 17 20 14 19 7 4

Output 12 :

201 178 155 140 129 122 111 107 92 81 64 44 30 11 4

Solution:

n=int(input())

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

l1=[]

for i in range(n):

    l2=l[i:]

    l1.append(sum(l2))

print(\*l1)

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

**Ques:104(137)**

Given a number N and an array of N elements, print the prefix sum array.

Input Size : N <= 100000

Sample Testcase :

INPUT

4

2 4 4 2

OUTPUT

2 6 10 12

Test Cases:

Input 1 :

16

24 13 9 9 6 22 16 15 1 6 23 4 8 4 23 8

Output 1 :

24 37 46 55 61 83 99 114 115 121 144 148 156 160 183 191

Input 2 :

18

12 16 6 5 24 2 15 20 13 12 9 24 20 7 6 3 23 17

Output 2 :

12 28 34 39 63 65 80 100 113 125 134 158 178 185 191 194 217 234

Input 3 :

13

8 11 5 18 12 1 23 2 10 8 19 10 24

Output 3 :

8 19 24 42 54 55 78 80 90 98 117 127 151

Input 4 :

14

10 6 12 3 20 11 11 21 13 23 3 16 13 23

Output 4 :

10 16 28 31 51 62 73 94 107 130 133 149 162 185

Input 5 :

17

5 8 23 23 9 21 19 24 14 7 8 6 22 17 13 13 8

Output 5 :

5 13 36 59 68 89 108 132 146 153 161 167 189 206 219 232 240

Input 6 :

13

14 19 24 13 1 10 8 1 18 18 8 1 11

Output 6 :

14 33 57 70 71 81 89 90 108 126 134 135 146

Input 7 :

17

17 3 3 17 20 20 17 6 15 14 12 14 17 22 8 1 17

Output 7 :

17 20 23 40 60 80 97 103 118 132 144 158 175 197 205 206 223

Input 8 :

14

1 12 13 1 6 20 15 2 18 6 12 4 7 22

Output 8 :

1 13 26 27 33 53 68 70 88 94 106 110 117 139

Input 9 :

13

6 5 15 8 18 22 22 23 19 20 21 6 10

Output 9 :

6 11 26 34 52 74 96 119 138 158 179 185 195

Input 10 :

10

10 15 8 6 1 16 21 14 17 21

Output 10 :

10 25 33 39 40 56 77 91 108 129

Input 11 :

18

2 3 23 24 1 4 17 1 9 11 17 2 3 21 14 23 20 14

Output 11 :

2 5 28 52 53 57 74 75 84 95 112 114 117 138 152 175 195 209

Input 12 :

15

23 23 15 11 7 11 4 15 11 17 20 14 19 7 4

Output 12 :

23 46 61 72 79 90 94 109 120 137 157 171 190 197 201

Solution:

n=int(input())

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

l1=[]

for i in range(n):

    l2=l[:i+1]

    l1.append(sum(l2))

print(\*l1)

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

**Ques:105(155)**

Given a number N and an array of N integers, predict if the product of all elements would be even or odd(actual multiplication may lead to overflows ai <= 100000000).If there is only one element present in the array print whether that number is odd or even.

Input Size : N <= 100000

Sample Testcase :

INPUT

4

2 4 4 2

OUTPUT

Even

Test Cases:

Input 1 :

16

24 13 9 9 6 22 16 15 1 6 23 4 8 4 23 8

Output 1 :

even

Input 2 :

18

12 16 6 5 24 2 15 20 13 12 9 24 20 7 6 3 23 17

Output 2 :

even

Input 3 :

13

8 11 5 18 12 1 23 2 10 8 19 10 24

Output 3 :

even

Input 4 :

14

10 6 12 3 20 11 11 21 13 23 3 16 13 23

Output 4 :

even

Input 5 :

17

5 8 23 23 9 21 19 24 14 7 8 6 22 17 13 13 8

Output 5 :

even

Input 6 :

13

14 19 24 13 1 10 8 1 18 18 8 1 11

Output 6 :

even

Input 7 :

17

17 3 3 17 20 20 17 6 15 14 12 14 17 22 8 1 17

Output 7 :

even

Input 8 :

14

1 12 13 1 6 20 15 2 18 6 12 4 7 22

Output 8 :

even

Input 9 :

13

6 5 15 8 18 22 22 23 19 20 21 6 10

Output 9 :

even

Input 10 :

10

10 15 8 6 1 16 21 14 17 21

Output 10 :

even

Input 11 :

18

2 3 23 24 1 4 17 1 9 11 17 2 3 21 14 23 20 14

Output 11 :

even

Input 12 :

15

23 23 15 11 7 11 5 15 11 17 21 15 19 7 41

Output 12 :

odd

Solution:

n=input()

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

p=1

for i in l:

    p=p\*i

if(p%2==0):

    print('even')

else:

    print('odd')

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

**Ques:106(153)**

Given 2 numbers N,K and an array of N integers, print all numbers in sorted order if it has been repeated less than K times.

Input Size : N,K <= 100000

Sample Testcase :

INPUT

4 2

2 4 4 1

OUTPUT

1 2

Test Cases:

Input 1 :

18 3

21 11 12 5 18 19 18 2 19 9 5 19 21 10 6 24 10 23

Output 1 :

2 5 5 6 9 10 10 11 12 18 18 21 21 23 24

Input 2 :

14 4

11 13 4 22 19 6 24 11 9 5 5 7 14 6

Output 2 :

4 5 5 6 6 7 9 11 11 13 14 19 22 24

Input 3 :

10 4

4 5 4 3 5 13 4 18 10 3

Output 3 :

3 3 4 4 4 5 5 10 13 18

Input 4 :

16 2

18 10 24 7 18 7 19 1 3 17 19 7 20 24 18 3

Output 4 :

1 10 17 20

Input 5 :

12 3

21 18 10 23 3 7 23 4 23 5 15 7

Output 5 :

3 4 5 7 7 10 15 18 21

Input 6 :

14 2

21 15 23 1  14 10 16 9 7 13 16 22 8 22

Output 6 :

1 7 8 9 10 13 14 15 21 23

Input 7 :

13 2

20 18 18 19 2 15 12 16 23 7 17 14 16

Output 7 :

2 7 12 14 15 17 19 20 23

Input 8 :

19 2

3 23 7 8 22 1 22 4 19 13 7 13 13 10 21 20 9 4 2

Output 8 :

1 2 3 8 9 10 19 20 21 23

Input 9 :

14 3

19 3 16 23 16 21 4 13 10 13 19 7 15 21

Output 9 :

3 4 7 10 13 13 15 16 16 19 19 21 21 23

Input 10 :

15 2

4 7 3 10 22 19 18 19 10 13 1 20 8 9 5

Output 10 :

1 3 4 5 7 8 9 13 18 20 22

Input 11 :

14 4

6 24 15 1 2 12 5 13 24 7 15 6 6 14

Output 11 :

1 2 5 6 6 6 7 12 13 14 15 15 24 24

Input 12 :

11 2

11 10 1 12 16 14 20 12 1 7 11

Output 12 :

7 10 14 16 20

Solution:

n,k=map(int,(input().split()))

a=list(map(int,(input().split())))

l=[]

for i in a:

    if(a.count(i)<k):

        l.append(i)

l.sort()

print(\*l)

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

**Ques:107(76)**

Given a number N and an array of N elements, every number is repeated except for one. Print that one number.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

10

1 2 3 2 3 3 2 5 5 2

OUTPUT

1

Test Cases:

Input 1 :

15

10 12 16 15 8 13 12 24 20 15 13 24 10 8 16

Output 1 :

20

Input 2 :

25

22 19 11 3 18 1 20 11 3 20 12 21 23 19 22 5 5 6 8 21 12 23 8 1 18

Output 2 :

6

Input 3 :

23

7 8 21 7 21 15 1 17 2 10 24 20 12 8 1 2 24 3 10 20 12 3 17

Output 3 :

15

Input 4 :

27

8 1 15 11 2 2 20 3 5 23 24 23 5 12 11 16 15 12 1 9 3 7 24 9 8 16 20

Output 4 :

7

Input 5 :

17

1 6 8 19 13 14 17 14 17 11 19 10 10 13 6 1 11

Output 5 :

8

Input 6 :

21

6 18 11 3 20 18 22 13 6 19 14 11 16 2 14 20 19 13 3 16 22

Output 6 :

2

Input 7 :

19

14 14 23 23 13 22 7 7 13 11 18 2 8 2 1 11 8 1 22

Output 7 :

18

Input 8 :

17

10 4 10 15 14 15 22 19 5 14 5 4 19 22 11 20 20

Output 8 :

11

Input 9 :

15

24 4 4 5 17 22 5 23 11 24 11 23 20 17 22

Output 9 :

20

Input 10 :

17

21 15 14 18 24 10 18 2 2 24 4 14 15 21 5 5 4

Output 10 :

10

Input 11 :

29

21 1 5 4 16 16 20 1 23 9 8 15 19 9 24 3 3 8 19 17 5 20 21 13 4 15 13 17 23

Output 11 :

24

Input 12 :

15

13 13 1 21 5 21 8 8 5 11 14 16 14 11 16

Output 12 :

1

Solution:

n=int(input())

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

for i in l:

    if(l.count(i)==1):

        print(i)

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

**Ques:108(127)**

Given a number N and a number K, check if it has all digits from 0 to k in it.

Input Size : N <= 100000

Sample Testcase :

INPUT

1234034 4

OUTPUT

Yes

Test Cases:

Input 1 :

74143 7

Output 1 :

No

Input 2 :

81163 6

Output 2 :

No

Input 3 :

83021 3

Output 3 :

Yes

Input 4 :

88957 8

Output 4 :

No

Input 5 :

22256 7

Output 5 :

No

Input 6 :

67581 7

Output 6 :

No

Input 7 :

71019 6

Output 7 :

No

Input 8 :

36082 5

Output 8 :

No

Input 9 :

12807 2

Output 9 :

Yes

Input 10 :

82108 2

Output 10 :

Yes

Input 11 :

49773 8

Output 11 :

No

Input 12 :

71070 7

Output 12 :

No

Solution:

n,k=map(int,input().split())

l=[]

for i in range(k+1):

    l.append(i)

a=list(map(int,str(n)))

l1=[]

for j in a:

    if j in l:

        if j not in l1:

            l1.append(j)

if(len(l)==len(l1)):

  print('yes')

else:

    print('no')

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

**Ques:109(343)**

Given k sorted arrays of possibly different sizes, merge them and print the sorted output.

Input Size : N<=100

Example:

INPUT

k = 3

1 3

2 4 6

0 9 10 11

OUTPUT

0 1 2 3 4 6 9 10 11

Test Cases:

Input 1 :

5

11344 19098 46260 51460 84862 99727

5718 12415 37366 59410 73666

493 6800 14162

6472 41759 46966 63766

4527 5014 19316 20718

Output 1 :

493 4527 5014 5718 6472 6800 11344 12415 14162 19098 19316 20718 37366 41759 46260 46966 51460 59410 63766 73666 84862 99727

Input 2 :

8

15264 32601 39364 46920

20500 53151 55978 88569 99093

23417 63165 63452

3981 28214 77375 89233

7210 16373 46839 87193

23188 32743 36743 64472 72112 83962

29525 56442 57830 82524

9611 11007 22304 33026 91758

Output 2 :

3981 7210 9611 11007 15264 16373 20500 22304 23188 23417 28214 29525 32601 32743 33026 36743 39364 46839 46920 53151 55978 56442 57830 63165 63452 64472 72112 77375 82524 83962 87193 88569 89233 91758 99093

Input 3 :

5

2461 22602 64288 97898

13591 25509 26041 51032 55373 86737

36493 40903 68704 85422

29935 53162 61629 84015

7923 16407 33706 37729 47950 52714

Output 3 :

2461 7923 13591 16407 22602 25509 26041 29935 33706 36493 37729 40903 47950 51032 52714 53162 55373 61629 64288 68704 84015 85422 86737 97898

Input 4 :

8

11373 56790 66602 93642 98068

22121 36781 48946 71808 95302

7596 14935 89575

25574 30962 49271 63359 67526

28648 54182 66715

1346 3386 9853 45273 97448

54531 92458 92997

2780 3118 10358 15539 40972 63607

Output 4 :

1346 2780 3118 3386 7596 9853 10358 11373 14935 15539 22121 25574 28648 30962 36781 40972 45273 48946 49271 54182 54531 56790 63359 63607 66602 66715 67526 71808 89575 92458 92997 93642 95302 97448 98068

Input 5 :

6

799 36562 67887 72173

1101 11424 46022 47323 61971 98161

24334 62505 90755

6964 23253 27561 72816

56216 59550 84367 85177 91155

39626 57475 66310 75032

Output 5 :

799 1101 6964 11424 23253 24334 27561 36562 39626 46022 47323 56216 57475 59550 61971 62505 66310 67887 72173 72816 75032 84367 85177 90755 91155 98161

Input 6 :

5

20786 48359 60096

45832 92474 93085 94311

17075 28295 28875 69615

4127 27599 67556 93841

13602 35704 52949

Output 6 :

4127 13602 17075 20786 27599 28295 28875 35704 45832 48359 52949 60096 67556 69615 92474 93085 93841 94311

Input 7 :

5

4126 10348 14388 60344 67078 92722

5613 14533 48687 91583

2895 6243 45083 51387 62651

6248 39588 59126 59700 78733 87405

9248 61183 68561 87459

Output 7 :

2895 4126 5613 6243 6248 9248 10348 14388 14533 39588 45083 48687 51387 59126 59700 60344 61183 62651 67078 68561 78733 87405 87459 91583 92722

Input 8 :

9

11064 21674 22014 31156 87501 92408

36941 45747 81146

6224 81489 92989

22192 49949 79502 92453

6280 40902 47458 48102

14430 19655 20707 27560 50774 78803

12714 23064 61034 93151

1341 30324 38811 51677 98806

26624 28271 60132 95628

Output 8 :

1341 6224 6280 11064 12714 14430 19655 20707 21674 22014 22192 23064 26624 27560 28271 30324 31156 36941 38811 40902 45747 47458 48102 49949 50774 51677 60132 61034 78803 79502 81146 81489 87501 92408 92453 92989 93151 95628 98806

Input 9 :

7

13024 13417 53134 61284

7301 8258 30872 47446 48242

8566 35399 56309 85983 94737

35566 79353 96510

7605 11476 27389 44621 71331 88345

14945 38291 95316 97815 98995

4942 20574 31505 38534 76768 88185

Output 9 :

4942 7301 7605 8258 8566 11476 13024 13417 14945 20574 27389 30872 31505 35399 35566 38291 38534 44621 47446 48242 53134 56309 61284 71331 76768 79353 85983 88185 88345 94737 95316 96510 97815 98995

Input 10 :

8

18192 84335 88701

6859 18040 18876 37794 66975 91392

63545 93946 98016

24876 48631 54250 84741

6875 12094 39549 55630

5278 61905 71961 89905

33098 48790 71295 87989

6729 63905 75758

Output 10 :

5278 6729 6859 6875 12094 18040 18192 18876 24876 33098 37794 39549 48631 48790 54250 55630 61905 63545 63905 66975 71295 71961 75758 84335 84741 87989 88701 89905 91392 93946 98016

Input 11 :

7

31233 75760 82843

16926 54443 90435 92164 96598

9788 46668 53845 58675 79596 79630

35370 57927 91189

15658 72777 90705 91895 93414

24093 31046 92801 99627

4137 41474 50368 61347 91626

Output 11 :

4137 9788 15658 16926 24093 31046 31233 35370 41474 46668 50368 53845 54443 57927 58675 61347 72777 75760 79596 79630 82843 90435 90705 91189 91626 91895 92164 92801 93414 96598 99627

Input 12 :

7

28190 33533 35466 36167 66006 87368

12370 25428 30245 48327 74813

46985 80637 96367

36271 55587 56455 59198 78551 78927

15677 28591 45162 99851

32296 36011 88461

28699 43625 52495 66257 97274

Output 12 :

12370 15677 25428 28190 28591 28699 30245 32296 33533 35466 36011 36167 36271 43625 45162 46985 48327 52495 55587 56455 59198 66006 66257 74813 78551 78927 80637 87368 88461 96367 97274 99851

Solution:

import numpy as np

n=int(input())

new=[]

for i in range(n):

    l=list(map(int,input().split()))

    new.append(l)

a=np.concatenate(new,axis=None)

a.sort()

print(\*a)

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

**Ques:110(158)**

A number is given as input. Find the odd digits in the number, add them and find if the sum is odd or not. If even print E, if odd print O.

Input Size : N <= 10000000000

Sample Testcase :

INPUT

413

OUTPUT

E

Test Cases:

Input 1 :

236478310

Output 1 :

E

Input 2 :

662761026

Output 2 :

E

Input 3 :

161400221

Output 3 :

O

Input 4 :

759205695

Output 4 :

E

Input 5 :

534127464

Output 5 :

E

Input 6 :

664302943

Output 6 :

O

Input 7 :

211013360

Output 7 :

O

Input 8 :

1712534

Output 8 :

O

Input 9 :

657294275

Output 9 :

O

Input 10 :

917815580

Output 10 :

E

Input 11 :

492690048

Output 11 :

E

Input 12 :

398144696

Output 12 :

E

Solution:

import re

n=input()

a=re.findall(r'\d',n)

d=list(map(int,a))

sum=0

for i in d:

    sum=sum+i

if(sum%2==0):

    print('E')

else:

    print('O')

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

Ques:111(513)

Iron Man wants to extract an infinity stone from a safe. The safe is protected by a password and Iron Man knows the clue to the password which is “sum one and two when sorted they are true”. Decode the clue from the test case and help Iron Man open the safe.

Input Description:

Size of the array followed by the elements of the array

Output Description:

Sum of 2 specific elements in the array.

Sample Input :

5

9 8 3 2 1

Sample Output :

3

Test Cases:

Input 1 :

5

32428 76919 97467 94881 48390

Output 1 :

80818

Input 2 :

9

17318 73614 97735 97995 72417 43988 49199 49298 45761

Output 2 :

61306

Input 3 :

9

95289 43936 27531 49440 90233 54288 1988 6722 16865

Output 3 :

8710

Input 4 :

7

29670 26808 66453 37789 57574 695 29853

Output 4 :

27503

Input 5 :

7

47405 48426 86478 31532 73085 91989 23573

Output 5 :

55105

Input 6 :

8

87956 48651 56837 45952 68507 11318 49797 50723

Output 6 :

57270

Input 7 :

7

82178 56080 83809 36430 79262 11963 29816

Output 7 :

41779

Input 8 :

7

99593 78949 63235 81955 82557 79198 7873

Output 8 :

71108

Input 9 :

6

26237 45715 15722 78766 31824 76434

Output 9 :

41959

Input 10 :

6

82681 90541 48470 27593 13715 89737

Output 10 :

41308

Input 11 :

9

34066 70732 46353 40729 63513 79305 91998 12018 53506

Output 11 :

46084

Input 12 :

9

96709 52014 57481 71243 41970 69605 47597 11418 80199

Output 12 :

53388

Solution:

n=int(input())

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

l.sort()

s=l[0]+l[1]

print(s)

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

**Ques:112(310)**

Given a number N, Print sum of every digit to the power of the weight of corresponding digit(Explanation : If the input is 12345 and then output calculated as (1^0)+(2^1)+(3^2)+(4^3)+(5^4)).

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

100

OUTPUT

1

Test Cases:

Input 1 :

17680

Output 1 :

556

Input 2 :

11180

Output 2 :

515

Input 3 :

59180

Output 3 :

523

Input 4 :

14542

Output 4 :

110

Input 5 :

72922

Output 5 :

108

Input 6 :

43724

Output 6 :

317

Input 7 :

55709

Output 7 :

6616

Input 8 :

84014

Output 8 :

262

Input 9 :

40409

Output 9 :

6578

Input 10 :

66022

Output 10 :

31

Input 11 :

85869

Output 11 :

6847

Input 12 :

64184

Output 12 :

774

Solution:

n=int(input())

t=n

c=0

while(n>0):

    r=n%10

    c=c+1

    n=n//10

i=1

s=0

while(t>0):

    f=t%10

    s=s+f\*\*(c-i)

     t=t//10

    i=i+1

print(s)

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

**Ques:113(599)**

Rahul was learning about numbers in list. He came across one word ground of a number.

A ground of a number is defined as the number which is just smaller or equal to the number given to you.Hence he started solving some assignments related to it. He got struck in some questions. Your task is to help him.

O(n) time complexity

O(n) Auxilary space

Input Description:

First line contains two numbers ‘n’ denoting number of integers and ‘k’ whose ground is to be check. Next line contains n space separated numbers.

Output Description:

Print the index of val.Print -1 if equal or near exqual number

Sample Input :

7 3

1 2 3 4 5 6 7

Sample Output :

2

Test Cases:

Input 1 :

8.0 8.0

6 7 1 10 13 10 2 14

Output 1 :

-1

Input 2 :

11.0 6.0

4 11 2 6 13 10 2 3 4 7 8

Output 2 :

3

Input 3 :

5.0 2.0

4 12 12 7 12

Output 3 :

-1

Input 4 :

12.0 4.0

11 12 8 5 10 12 9 8 9 6 6 8

Output 4 :

-1

Input 5 :

5.0 1.0

11 13 13 3 12

Output 5 :

-1

Input 6 :

12.0 6.0

9 8 6 6 10 9 11 1 4 3 8 11

Output 6 :

2

Input 7 :

10.0 14.0

8 3 12 11 6 1 3 1 11 13

Output 7 :

-1

Input 8 :

10.0 12.0

6 8 5 4 2 8 9 9 4 5

Output 8 :

-1

Input 9 :

6.0 14.0

7 7 12 6 13 3

Output 9 :

-1

Input 10 :

11.0 9.0

5 9 14 12 11 8 1 3 14 10 7

Output 10 :

1

Input 11 :

14.0 4.0

13 13 3 10 4 4 6 3 9 2 5 12 13 10

Output 11 :

4

Input 12 :

13.0 4.0

2 10 13 4 7 13 7 5 11 4 6 2 11

Output 12 :

3

Solution:

import math

n,k=map(float,(input().split()))

a=math.floor(k)

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

for i in l:

    if(i==a):

        print(l.index(i))

        break

else:

    print(-1)

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

**Ques:114(209)**

Given a number N, print the first N Catalan numbers.

Input Size : N <= 100

Sample Testcase :

INPUT

1

OUTPUT

1

Test Cases:

Input 1 :

1

Output 1 :

1

Input 2 :

2

Output 2 :

1 1

Input 3 :

3

Output 3 :

1 1 2

Input 4 :

4

Output 4 :

1 1 2 5

Input 5 :

5

Output 5 :

1 1 2 5 14

Input 6 :

6

Output 6 :

1 1 2 5 14 42

Input 7 :

7

Output 7 :

1 1 2 5 14 42 132

Input 8 :

8

Output 8 :

1 1 2 5 14 42 132 429

Input 9 :

9

Output 9 :

1 1 2 5 14 42 132 429 1430

Input 10 :

10

Output 10 :

1 1 2 5 14 42 132 429 1430 4862

Input 11 :

11

Output 11 :

1 1 2 5 14 42 132 429 1430 4862 16796

Input 12 :

12

Output 12 :

1 1 2 5 14 42 132 429 1430 4862 16796 58786

Solution:

def catalan(n):

    if n <= 1 :

        return 1

    s = 0

    for i in range(n):

        s += catalan(i) \* catalan(n-i-1)

    return s

n=int(input())

for i in range(n):

    print(catalan(i),end=" ")

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

**Ques:115(574)**

You are given a number n,an array of n numbers ranging from 1 to n. Out of which one number is missing. Your task is to print that missing number.

Input Description:

You are given a number ‘n’.

Output Description:

Print the missing number.

Sample Input :

5

1 3 5 2

Sample Output :

4

Test Cases:

Input 1 :

13

5 9 8 2 11 4 6 3 10 7 12 1

Output 1 :

13

Input 2 :

10

1 6 2 3 7 5 10 4 9

Output 2 :

8

Input 3 :

10

5 8 9 1 7 3 4 10 2

Output 3 :

6

Input 4 :

11

7 3 11 1 10 8 6 4 9 2

Output 4 :

5

Input 5 :

8

4 5 1 3 6 7 8

Output 5 :

2

Input 6 :

11

1 11 9 10 3 4 2 8 6 5

Output 6 :

7

Input 7 :

13

3 6 2 11 12 5 7 8 1 9 13 10

Output 7 :

4

Input 8 :

9

5 4 2 6 1 3 7 9

Output 8 :

8

Input 9 :

6

4 1 5 2 6

Output 9 :

3

Input 10 :

6

2 6 5 3 1

Output 10 :

4

Input 11 :

8

8 7 3 1 5 4 6

Output 11 :

2

Input 12 :

8

2 5 4 7 6 8 1

Output 12 :

3

Solution:

n=int(input())

a=list(map(int,input().split()))

total=n\*(n+1)/2

s=sum(a)

miss=total-s

print(int(miss))

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

**Ques:116(240)**

Given a number n followed by n numbers. Find the numbers which are equal to their index value and print them in sorted order. If no such numbers are present print '-1' without quotes.

Input Size : 1 <= n <= 100000

Sample Testcases :

INPUT

6

6 7 3 3 4 5

OUTPUT

3 4 5

Test Cases:

Input 1 :

13

5 9 8 2 11 4 6 3 10 7 12 1

Output 1 :

6

Input 2 :

10

1 6 2 3 7 5 10 4 9

Output 2 :

2 3 5

Input 3 :

10

5 8 9 1 7 3 4 10 2

Output 3 :

-1

Input 4 :

11

7 3 11 1 10 8 6 4 9 2

Output 4 :

6

Input 5 :

8

4 5 1 3 6 7 8

Output 5 :

3

Input 6 :

11

1 11 9 10 3 4 2 8 6 5

Output 6 :

-1

Input 7 :

13

3 6 2 11 12 5 7 8 1 9 13 10

Output 7 :

2 5 9

Input 8 :

9

5 4 2 6 1 3 7 9

Output 8 :

2

Input 9 :

6

4 1 5 2 6

Output 9 :

1

Input 10 :

6

2 6 5 3 1

Output 10 :

3

Input 11 :

8

8 7 3 1 5 4 6

Output 11 :

6

Input 12 :

8

2 5 4 7 6 8 1

Output 12 :

-1

Solution:

n=int(input())

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

d=[]

for i in l:

    a=l.index(i)

    b=i

    if(a==b):

        d.append(b)

if(len(d)==0):

    print(-1)

else:

    print(\*d)

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

**Ques:117(239)**

Given a number N followed by N numbers. Out of these N numbers some of them are repeated. Write a program to find the number which is repeated and print the repeated numbers in sorted order. If no numbers are repeated print "unique".

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

7

1 2 3 2 3 4 3

OUTPUT

2 3

Test Cases:

Input 1 :

14

9 7 6 5 15 1 3 18 4 11 12 19 13 2

Output 1 :

unique

Input 2 :

11

7 1 1 2 9 6 4 3 5 6 7

Output 2 :

1 6 7

Input 3 :

9

1 5 7 2 2 5 3 7 3

Output 3 :

2 3 5 7

Input 4 :

6

2 1 1 1 2 2

Output 4 :

1 2

Input 5 :

13

5 8 2 11 9 7 9 12 8 2 1 8 1

Output 5 :

1 2 8 9

Input 6 :

14

9 11 5 6 5 4 1 8 12 12 5 5 8 13

Output 6 :

5 8 12

Input 7 :

5

2 2 4 3 2

Output 7 :

2

Input 8 :

10

2 1 2 7 6 7 9 1 4 9

Output 8 :

1 2 7 9

Input 9 :

6

3 4 4 3 2 5

Output 9 :

3 4

Input 10 :

9

5 2 3 6 8 6 3 4 8

Output 10 :

3 6 8

Input 11 :

8

7 3 7 7 1 7 4 1

Output 11 :

1 7

Input 12 :

10

3 7 7 5 5 1 8 4 9 3

Output 12 :

3 5 7

Solution:

n=int(input())

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

a=[]

for i in l:

    if(l.count(i)!=1):

        if i not in a:

            a.append(i)

if(len(a)==0):

    print('unique')

else:

    a.sort()

    print(\*a)

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

**Ques:118(238)**

Given a number N followed by N numbers. Find the largest number which can be formed from the given numbers and print it.

NOTE: Each number should be used exactly once.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

5

5 6 7 8 9

OUTPUT

98765

Test Cases:

Input 1 :

14

9 7 6 5 15 1 3 18 4 11 12 19 13 2

Output 1 :

19181513121197654321

Input 2 :

11

7 1 1 2 9 6 4 3 5 6 7

Output 2 :

97766543211

Input 3 :

9

1 5 7 2 2 5 3 7 3

Output 3 :

775533221

Input 4 :

6

2 1 1 1 2 2

Output 4 :

222111

Input 5 :

13

5 8 2 11 9 7 9 12 8 2 1 8 1

Output 5 :

121199888752211

Input 6 :

14

9 11 5 6 5 4 1 8 12 12 5 5 8 13

Output 6 :

131212119886555541

Input 7 :

5

2 2 4 3 2

Output 7 :

43222

Input 8 :

10

2 1 2 7 6 7 9 1 4 9

Output 8 :

9977642211

Input 9 :

6

3 4 4 3 2 5

Output 9 :

544332

Input 10 :

9

5 2 3 6 8 6 3 4 8

Output 10 :

886654332

Input 11 :

8

7 3 7 7 1 7 4 1

Output 11 :

77774311

Input 12 :

10

3 7 7 5 5 1 8 4 9 3

Output 12 :

9877554331

Solution:

n=int(input())

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

l.sort(reverse=True)

print(\*l,sep="")

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

**Ques:119(247)**

Given a string S of length N, write a program that would reverse every word in the string.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

Hello World

OUTPUT

olleH dlroW

Test Cases:

Input 1 :

mTeV DbPS MSm wBvbE t iKo

Output 1 :

VeTm SPbD mSM EbvBw t oKi

Input 2 :

ObGq ToSb j rFY Qit azhEh

Output 2 :

qGbO bSoT j YFr tiQ hEhza

Input 3 :

TkAOo x H O F Ewa D z

Output 3 :

oOAkT x H O F awE D z

Input 4 :

jIx dD DZy YdpiA AsHII KMh l z

Output 4 :

xIj Dd yZD AipdY IIHsA hMK l z

Input 5 :

f XXIFH fl p M gwmxO Qr

Output 5 :

f HFIXX lf p M Oxmwg rQ

Input 6 :

riaFs SAErJ at YZL uUQ qXmy Q Yq JVx

Output 6 :

sFair JrEAS ta LZY QUu ymXq Q qY xVJ

Input 7 :

V t a Ovxb KNxJ vTPwh GW Zhv iAWDu

Output 7 :

V t a bxvO JxNK hwPTv WG vhZ uDWAi

Input 8 :

mJHfB PL JXOt bwYBu XxaOF GAZix

Output 8 :

BfHJm LP tOXJ uBYwb FOaxX xiZAG

Input 9 :

Xmmw fqW Q m ANy

Output 9 :

wmmX Wqf Q m yNA

Input 10 :

LL VZ pgI V MwwF PQdhJ jLd

Output 10 :

LL ZV Igp V FwwM JhdQP dLj

Input 11 :

ee Qf Ld I uI AFIXh IJfTw Ger

Output 11 :

ee fQ dL I Iu hXIFA wTfJI reG

Input 12 :

th BYLbH zljC MwNJ DtfNO

Output 12 :

ht HbLYB Cjlz JNwM ONftD

Solution:

s=input()

a=s.split()

l=[]

for i in a:

    a=i[::-1]

    l.append(a)

print(\*l)

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

**Ques:120(248)**

Given 2 numbers N and M followed by 2 arrays A an B of sizes N and M. Check if array B is a subset of array A. if yes print 'yes' else print 'no'.

Input Size : 1 <= N <= 1000000

Sample Testcases :

INPUT

7 6

1 2 3 4 5 6 7

3 4 5 6 7 1

OUTPUT

Yes

Test Cases:

Input 1 :

9 9

80917 53116 49773 53963 26490 94739 73958 94724 14649

53963 92408 32075 14649 94724 26490 53116 63480 94739

Output 1 :

no

Input 2 :

12 10

90060 66644 17193 68160 52517 98677 14132 3783 91552 36688 82470 65742

87135 85682 11065 98677 82470 17620 52517 90060 66644 69345

Output 2 :

no

Input 3 :

14 7

90170 8145 66806 11249 75848 33272 94645 84900 91938 4486 31431 34105 24042 45495

75848 34105 91938 84900 94645 31431 66806

Output 3 :

yes

Input 4 :

13 6

683 74966 60127 46177 76339 51497 32351 66937 68671 76397 68866 95518 61315

68671 32351 66937 61032 74966 28691

Output 4 :

no

Input 5 :

10 11

92467 28009 99227 62340 96109 143 97398 13117 77546 87185

7576 75561 48814 69175 94017 50849 73060 40482 34770 42955 94786

Output 5 :

no

Input 6 :

10 8

36313 81216 87860 17943 13343 20930 33358 11804 96647 36195

36195 13343 20930 11804 36313 20780 33358 74010

Output 6 :

no

Input 7 :

11 10

73731 72704 7328 61475 70990 64574 65691 11432 93667 41203 4962

93667 61475 38364 8817 65691 73731 62043 43928 64574 7328

Output 7 :

no

Input 8 :

6 14

20772 41737 96434 77209 41706 62444

18942 19356 6074 85782 34903 62991 38068 32799 15776 59391 21101 3182 41238 61796

Output 8 :

no

Input 9 :

8 10

42736 10879 38980 16575 55782 56614 9523 44191

9381 78523 11794 78142 70812 76352 87482 24398 1782 54750

Output 9 :

no

Input 10 :

5 8

20301 73077 524 43514 41770

25584 23068 75008 47294 31493 47230 48517 11967

Output 10 :

no

Input 11 :

9 13

43438 76990 13328 67489 91029 70927 38521 95055 59209

95527 93836 69333 96933 11695 70519 67381 96526 30622 55810 40116 22021 89777

Output 11 :

no

Input 12 :

6 11

80649 16484 88250 28596 58943 31285

33065 64293 82651 24995 45630 58013 88140 69041 59473 58970 15105

Output 12 :

no

Solution:

n,m=map(int,input().split())

a=list(map(int,input().split()))

b=list(map(int,input().split()))

A=set(a)

B=set(b)

if(B.issubset(A)):

    print('yes')

else:

    print('no')

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

**Ques:121(393)**

Sensex has never been steady and stock values have been rising and falling in the past. Ram is a stock broker and on analysing the trend of change in sensex points, he decides that there was a period when the points was at its all time high and investing during that period would have earned him a huge profit in trade. Ram immediately regrets not having done any trade in that particular period. Find out which period in time was Ram referring to.

Constraints:

3 ≤ N ≤ 106

Input Description:

The size of array N, the number of periods Ram has assessed. The values of the array are the values of the sensex points.

Output Description:

Period in which the points was at its max value. The array index is the period value.

Sample Input :

9

1 15 25 45 42 21 17 12 11

Sample Output :

3

Test Cases:

Input 1 :

12

61168 15401 13956 50026 42732 37318 49926 47723 59423 16455 22304 1964

Output 1 :

0

Input 2 :

12

28355 66501 24448 20161 66306 2470 67046 23627 39214 95818 47518 78424

Output 2 :

9

Input 3 :

13

84378 32927 28711 43598 8370 53415 62569 49716 42474 53539 93946 73070 39985

Output 3 :

10

Input 4 :

7

73576 16966 90203 13503 62356 68231 46205

Output 4 :

2

Input 5 :

12

69114 52012 4012 70536 71619 28196 83892 87454 36434 31609 98922 30225

Output 5 :

10

Input 6 :

5

55258 33464 58321 96720 10552

Output 6 :

3

Input 7 :

10

5619 94806 13010 53405 13929 85067 61700 21868 51555 50105

Output 7 :

1

Input 8 :

6

19012 46728 97936 53450 18256 12172

Output 8 :

2

Input 9 :

6

26683 93212 35563 61759 63809 99175

Output 9 :

5

Input 10 :

14

20313 72141 71337 670 71896 60698 55372 46212 60369 69668 10817 15977 82322 67514

Output 10 :

12

Input 11 :

12

15657 21471 46124 93695 66550 56897 34454 90955 43645 65919 76350 36125

Output 11 :

3

Input 12 :

14

43065 42219 58316 3392 76086 23648 46542 76158 50852 19748 54726 75563 49050 89918

Output 12 :

13

Solution:

n=int(input())

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

a=max(l)

print(l.index(a))

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

**Ques:122(98)**

Given a number N, print the even factors of N.If the even factor does not exists for N print '-1'.

Input Size : 1 <= N <= 1000

Sample Testcase :

INPUT

8

OUTPUT

2 4 8

Test Cases:

Input 1 :

9

Output 1 :

-1

Input 2 :

18

Output 2 :

2 6 18

Input 3 :

25

Output 3 :

-1

Input 4 :

19

Output 4 :

-1

Input 5 :

11

Output 5 :

-1

Input 6 :

23

Output 6 :

-1

Input 7 :

11

Output 7 :

-1

Input 8 :

22

Output 8 :

2 22

Input 9 :

8

Output 9 :

2 4 8

Input 10 :

21

Output 10 :

-1

Input 11 :

15

Output 11 :

-1

Input 12 :

13

Output 12 :

-1

Solution:

a=int(input())

for i in range(1,a+1):

    if(a%i==0):

        if(i%2==0):

            print(i,end="")

if(i%2!=0):

    print('-1')

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

**Ques:123(226)**

Given a number N followed by N numbers(negative or positive) print the maximum sum of any subarray of the array.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

7

1 2 3 4 5 6 7

OUTPUT

28

Test Cases:

Input 1 :

4

7 6 -4 -8

Output 1 :

13

Input 2 :

4

-2 0 5 7

Output 2 :

12

Input 3 :

12

14 9 6 -15 11 -7 11 4 15 -1 -13 -19

Output 3 :

48

Input 4 :

9

13 -8 13 0 -17 9 -19 -3 -1

Output 4 :

18

Input 5 :

1

2

Output 5 :

2

Input 6 :

2

-8 5

Output 6 :

5

Input 7 :

13

-18 14 8 -22 10 15 18 -1 -8 -1 -14 14 -1

Output 7 :

43

Input 8 :

15

17 7 -2 3 3 -13 -4 -10 1 16 -22 -23 -11 -2 -12

Output 8 :

28

Input 9 :

12

4 12 -8 10 4 -10 -6 -13 -18 6 -9 -4

Output 9 :

22

Input 10 :

11

-3 7 2 -1 -7 2 -18 16 -7 -5 -6

Output 10 :

16

Input 11 :

13

-18 -3 -19 -7 -3 -19 8 10 6 -15 6 6 -4

Output 11 :

24

Input 12 :

22

-29 15 9 -25 -12 -24 21 23 -16 -28 -11 -3 3 -1 -24 8 -5 4 3 -27 26 3

Output 12 :

44

Solution:

n=int(input())

a=list(map(int,input().split()))

sublist=[]

for i in range(len(a)+1):

    for j in range(i+1,len(a)+1):

        s=a[i:j]

        sublist.append(sum(s))

print(max(sublist))

from sys import maxsize

def maxSubArraySum(a,size):

   max\_so\_far = -maxsize - 1

   max\_ending\_here = 0

   for i in range(0, size):

       max\_ending\_here = max\_ending\_here + a[i]

       if (max\_so\_far < max\_ending\_here):

           max\_so\_far = max\_ending\_here

        if max\_ending\_here < 0:

           max\_ending\_here = 0

   return max\_so\_far

n=int(input())

a=list(map(int,input().split()))

print(maxSubArraySum(a,n))

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

**Ques:124(237)**

Given a number N, check if the sum of the digits is a Palindrome. Print 'yes' or 'no' accordingly.

Input Size : 2 <= N <= 1000000000000000000

Sample Testcases :

INPUT

13

OUTPUT

Yes

Test Cases:

Input 1 :

241734772756

Output 1 :

yes

Input 2 :

633000434917

Output 2 :

no

Input 3 :

594038704573

Output 3 :

yes

Input 4 :

550658929124

Output 4 :

no

Input 5 :

615767276585

Output 5 :

no

Input 6 :

670230567571

Output 6 :

no

Input 7 :

299130892932

Output 7 :

no

Input 8 :

434108474358

Output 8 :

no

Input 9 :

395133863834

Output 9 :

no

Input 10 :

605451438838

Output 10 :

yes

Input 11 :

364181058007

Output 11 :

no

Input 12 :

202629199489

Output 12 :

no

Solution:

import re

s=input()

a=re.findall(r'\d',s)

d=list(map(int,a))

sum=0

for i in d:

    sum=sum+i

s=str(sum)

if(s==s[::-1]):

    print('yes')

else:

    print('no')

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

**Ques:125(249)**

Given two numbers N,K(N>=K) and an array of N elements, write a program to find the Kth largest element.

Input Size : 1 <= K <= N <= 100000

Sample Testcases :

INPUT

6 2

1 2 3 4 5 6

OUTPUT

5

Input 1 :

4 2

6 5 -8 0

Output 1 :

5

Input 2 :

12 11

1 13 -13 -2 5 -18 -15 -22 11 -4 6 -12

Output 2 :

-18

Input 3 :

5 2

-7 -4 3 -1 -3

Output 3 :

-1

Input 4 :

8 2

12 -12 -15 -10 -15 8 -16 8

Output 4 :

8

Input 5 :

5 4

-5 -3 5 5 -1

Output 5 :

-3

Input 6 :

2 1

7 -10

Output 6 :

7

Input 7 :

6 2

7 -8 -1 -13 -12 -12

Output 7 :

-1

Input 8 :

7 6

-2 6 7 -13 -16 1 9

Output 8 :

-13

Input 9 :

6 2

-5 -16 -2 -3 -1 -6

Output 9 :

-2

Input 10 :

6 1

-10 7 11 -15 3 -1

Output 10 :

11

Input 11 :

2 1

-3 -2

Output 11 :

-2

Input 12 :

7 4

9 1 -10 -14 2 -10 -17

Output 12 :

-10

Solution:

n,k=map(int,(input().split()))

a=list(map(int,input().split()))

a.sort(reverse=True)

print(a[k-1])

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

**Ques:126(279)**

Given a number N followed by an array of N integers, every element appears twice except for one. Find that single one and print it.

Input Size : N <= 100000

Sample Testcase :

INPUT

5

30 5 5 30 -45

OUTPUT

-45

Test Cases:

Input 1 :

9

6 -5 1 -10 1 -4 -4 6 -10

Output 1 :

-5

Input 2 :

7

-7 6 2 -7 -12 -12 6

Output 2 :

2

Input 3 :

5

3 -6 -6 -12 3

Output 3 :

-12

Input 4 :

5

3 -2 10 3 10

Output 4 :

-2

Input 5 :

9

10 -12 -2 13 10 -12 -11 -2 13

Output 5 :

-11

Input 6 :

7

-7 -7 -13 -13 13 13 -19

Output 6 :

-19

Input 7 :

3

-2 10 -2

Output 7 :

10

Input 8 :

7

9 -10 -18 -1 -18 -10 -1

Output 8 :

9

Input 9 :

7

-9 7 -13 -4 -9 -13 -4

Output 9 :

7

Input 10 :

5

0 -1 -14 0 -14

Output 10 :

-1

Input 11 :

11

-11 -3 2 2 17 -11 -1 10 17 10 -1

Output 11 :

-3

Input 12 :

5

-17 -17 10 10 -12

Output 12 :

-12

Solution:

n=int(input())

a=list(map(int,input().split()))

for i in a:

    if(a.count(i)==1):

        print(i)

        break

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

**Ques:127(298)**

Given a number N, print all prime numbers less than N(in ascending order).

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

10

OUTPUT

2 3 5 7

Test Cases:

Input 1 :

29

Output 1 :

2 3 5 7 11 13 17 19 23

Input 2 :

32

Output 2 :

2 3 5 7 11 13 17 19 23 29 31

Input 3 :

3

Output 3 :

2

Input 4 :

16

Output 4 :

2 3 5 7 11 13

Input 5 :

23

Output 5 :

2 3 5 7 11 13 17 19

Input 6 :

20

Output 6 :

2 3 5 7 11 13 17 19

Input 7 :

27

Output 7 :

2 3 5 7 11 13 17 19 23

Input 8 :

7

Output 8 :

2 3 5

Input 9 :

15

Output 9 :

2 3 5 7 11 13

Input 10 :

31

Output 10 :

2 3 5 7 11 13 17 19 23 29

Input 11 :

23

Output 11 :

2 3 5 7 11 13 17 19

Input 12 :

30

Output 12 :

2 3 5 7 11 13 17 19 23 29

Solution:

r=int(input())

l=[]

for a in range(2,r):

    k=0

    for i in range(2,a//2+1):

        if(a%i==0):

            k=k+1

    if(k<=0):

        l.append(a)

print(\*l)

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

**Ques:128(318)**

Given a number N followed by N numbers. Keep the count of each number and print the maximum repeating number.

Input Size : N <= 100000

Sample Testcase :

INPUT

5

15 5 -20 -20 -45

OUTPUT

-20

Test Cases:

Input 1 :

17

4 7 6 3 7 5 7 3 6 5 6 10 5 1 7 3 5

Output 1 :

7

Input 2 :

9

1 4 4 10 10 3 5 6 6

Output 2 :

4

Input 3 :

11

6 2 4 5 5 5 5 3 2 5 1

Output 3 :

5

Input 4 :

12

7 4 7 10 9 4 5 5 10 10 3 1

Output 4 :

10

Input 5 :

5

4 6 7 4 2

Output 5 :

4

Input 6 :

5

1 1 7 4 2

Output 6 :

1

Input 7 :

15

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

Output 7 :

1

Input 8 :

8

1 3 7 4 8 2 4 2

Output 8 :

4

Input 9 :

5

5 5 1 10 7

Output 9 :

5

Input 10 :

7

5 4 8 8 4 8 7

Output 10 :

8

Input 11 :

12

4 2 9 3 8 6 7 5 3 2 5 3

Output 11 :

3

Input 12 :

18

4 8 8 9 9 1 9 8 6 2 4 8 2 1 2 10 6 9

Output 12 :

8

Solution:

n=int(input())

a=list(map(int,input().split()))

for i in a:

    if(a.count(i)!=1):

        print(i)

        Break

import collections

n=int(input())

a=list(map(int,input().split()))

freq = collections.Counter(a)

print(freq.most\_common()[0][0])

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

**Ques:129(290)**

Given a number N followed by N numbers.Find the smallest number and largest number and print both the indices(1 based indexing).

Input Size : N <= 100000

Sample Testcase :

INPUT

5

1 2 3 4 5

OUTPUT

1 5

Test Cases:

Input 1 :

7

231 203 847 222 117 447 401

Output 1 :

5 3

Input 2 :

14

258 951 43 468 726 340 450 641 128 508 696 11 844 354

Output 2 :

12 2

Input 3 :

17

858 12 334 738 903 350 437 312 983 162 527 768 652 660 980 756 868

Output 3 :

2 9

Input 4 :

18

510 16 831 61 191 739 704 2 469 914 194 765 267 499 166 735 308 5

Output 4 :

8 10

Input 5 :

10

163 566 91 384 588 431 587 933 64 510

Output 5 :

9 8

Input 6 :

16

291 71 198 371 550 379 454 117 260 218 519 221 508 147 899 834

Output 6 :

2 15

Input 7 :

11

610 409 44 410 1000 47 657 873 300 363 480

Output 7 :

3 5

Input 8 :

6

876 230 119 440 693 983

Output 8 :

3 6

Input 9 :

15

532 209 449 384 576 471 460 56 354 511 34 465 63 217 783

Output 9 :

11 15

Input 10 :

12

821 162 963 615 787 144 833 451 351 654 432 85

Output 10 :

12 3

Input 11 :

7

949 135 832 339 988 644 892

Output 11 :

2 5

Input 12 :

13

665 841 904 775 870 832 992 777 809 484 941 319 707

Output 12 :

12 7

Solution:

n=int(input())

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

s=min(l)

d=max(l)

print((l.index(s)+1),end=" ")

print((l.index(d)+1))

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

**Ques:130(607)**

Ramesh is given a special series to print,as he has some other work to do.Help ramesh in printing the series.

Note:Observe the series very keenly in examples

Input Description:

You are given a number ‘n’.

Output Description:

Print the n term of series.

Sample Input :

6

Sample Output :

1 6 120

Test Cases:

Input 1 :

10

Output 1 :

1 6 120 5040 362880

Input 2 :

13

Output 2 :

1 6 120 5040 362880 39916800 6227020800

Input 3 :

8

Output 3 :

1 6 120 5040

Input 4 :

3

Output 4 :

1 6

Input 5 :

4

Output 5 :

1 6

Input 6 :

11

Output 6 :

1 6 120 5040 362880 39916800

Input 7 :

6

Output 7 :

1 6 120

Input 8 :

13

Output 8 :

1 6 120 5040 362880 39916800 6227020800

Input 9 :

12

Output 9 :

1 6 120 5040 362880 39916800

Input 10 :

9

Output 10 :

1 6 120 5040 362880

Input 11 :

10

Output 11 :

1 6 120 5040 362880

Input 12 :

5

Output 12 :

1 6 120

Solution:

import math

n=int(input())

l=[]

for i in range(n+1):

    if(i%2!=0):

        a=math.factorial(i)

        l.append(a)

print(\*l)

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

**Ques:131(287)**

Given 2 numbers N and K and followed by an array of N integers. The given element K is removed from the array and new array is printed. If after removing every occurance of K the array becomes empty, print 'empty' without quotes.

Example:

Input: {10,10,20,30,76}, K=10

Output: {20,20,76}

Input Size : N <= 100000

Sample Testcase :

INPUT

5 10

10 10 20 30 76

OUTPUT

20 30 76

Test Cases:

Input 1 :

10 11

11 6 14 7 11 9 13 5 7 10

Output 1 :

6 14 7 9 13 5 7 10

Input 2 :

6 3

8 11 15 3 3 1

Output 2 :

8 11 15 1

Input 3 :

4 2

2 2 2 2

Output 3 :

empty

Input 4 :

5 1

14 5 12 12 8

Output 4 :

14 5 12 12 8

Input 5 :

14 6

6 12 6 3 11 13 8 6 8 15 3 13 1 11

Output 5 :

12 3 11 13 8 8 15 3 13 1 11

Input 6 :

6 3

11 4 12 6 1 5

Output 6 :

11 4 12 6 1 5

Input 7 :

9 15

7 15 7 6 2 15 15 12 3

Output 7 :

7 7 6 2 12 3

Input 8 :

14 1

1 1 7 8 3 10 9 2 4 8 15 15 12 2

Output 8 :

7 8 3 10 9 2 4 8 15 15 12 2

Input 9 :

10 12

11 6 5 5 5 10 12 4 13 3

Output 9 :

11 6 5 5 5 10 4 13 3

Input 10 :

2 1

5 6

Output 10 :

5 6

Input 11 :

12 13

7 12 13 10 7 5 14 8 13 1 13 3

Output 11 :

7 12 10 7 5 14 8 1 3

Input 12 :

8 5

3 14 10 11 15 2 13 15

Output 12 :

3 14 10 11 15 2 13 15

Solution:

a,b=map(int,input().split())

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

c=[]

d=[]

for i in l:

    if(i==b):

        d.append(i)

    else:

        c.append(i)

if(len(c)==0):

    print('empty')

else:

    print(\*c)

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

**Ques:132(242)**

Given a number N followed by N numbers. All the numbers in the given input appear twice except for one number(ie one number appears only once in the given input). Find the number which appears only once.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

5

1 2 1 3 2

OUTPUT

3

Test Cases:

Input 1 :

9

6 -5 1 -10 1 -4 -4 6 -10

Output 1 :

-5

Input 2 :

7

-7 6 2 -7 -12 -12 6

Output 2 :

2

Input 3 :

5

3 -6 -6 -12 3

Output 3 :

-12

Input 4 :

5

3 -2 10 3 10

Output 4 :

-2

Input 5 :

9

10 -12 -2 13 10 -12 -11 -2 13

Output 5 :

-11

Input 6 :

7

-7 -7 -13 -13 13 13 -19

Output 6 :

-19

Input 7 :

3

-2 10 -2

Output 7 :

10

Input 8 :

7

9 -10 -18 -1 -18 -10 -1

Output 8 :

9

Input 9 :

7

-9 7 -13 -4 -9 -13 -4

Output 9 :

7

Input 10 :

5

0 -1 -14 0 -14

Output 10 :

-1

Input 11 :

11

-11 -3 2 2 17 -11 -1 10 17 10 -1

Output 11 :

-3

Input 12 :

5

-17 -17 10 10 -12

Output 12 :

-12

Solution:

n=int(input())

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

for i in l:

    if(l.count(i)==1):

        print(i)

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

Ques:133(321)

Given a string S convert each characters of the string into ASCII values and print the sum of the numbers.

Input Size : |s| <= 100000

Sample Testcase :

INPUT

guvi

OUTPUT

443

Test Cases:

Input 1 :

NvILfpLDNWgYY

Output 1 :

1149

Input 2 :

SFJBpXFe

Output 2 :

664

Input 3 :

QwoHjbmVlBjR

Output 3 :

1144

Input 4 :

mMNvrlaizAS

Output 4 :

1076

Input 5 :

aMbwZ

Output 5 :

481

Input 6 :

OImJMfpaYioZg

Output 6 :

1221

Input 7 :

QgaLvnGSsGpk

Output 7 :

1144

Input 8 :

BophwHKFrkWlgE

Output 8 :

1317

Input 9 :

TEfJiALqLK

Output 9 :

839

Input 10 :

MUumZBbswDJPyNR

Output 10 :

1379

Input 11 :

CmzLTTulpSlnDNc

Output 11 :

1425

Input 12 :

kdIEh

Output 12 :

453

Solution:

import re

s=input()

a=re.findall(r'\w',s)

d=list(map(str,a))

sum=0

for i in d:

    b=ord(i)

    sum=sum+b

print(sum)

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

**Ques:134(250)**

Given a string S of length N, find whether the given string is a palindrome using stack or linked list and print 'yes' otherwise print 'no'.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

8

GuviGeek

OUTPUT

No

Test Cases:

Input 1 :

10

tBMxaaxMBt

Output 1 :

Yes

Input 2 :

20

TKdXrpXqKNTKdXrpXqKN

Output 2 :

No

Input 3 :

14

FNBSlWZFNBSlWZ

Output 3 :

No

Input 4 :

8

YoSrYoSr

Output 4 :

No

Input 5 :

28

EmehuublvcAwwzEmehuublvcAwwz

Output 5 :

No

Input 6 :

6

loNloN

Output 6 :

No

Input 7 :

30

IlMuKIvHbPjSKDeeDKSjPbHvIKuMlI

Output 7 :

Yes

Input 8 :

18

pyybTkNbopyybTkNbo

Output 8 :

No

Input 9 :

22

rYYibKOdCiHrYYibKOdCiH

Output 9 :

No

Input 10 :

22

tgZRbnVADxCCxDAVnbRZgt

Output 10 :

Yes

Input 11 :

28

PrLHaUtOpbhfoUPrLHaUtOpbhfoU

Output 11 :

No

Input 12 :

8

xUKxxUKx

Output 12 :

No

Solution:

n = int(input())

s=input()

s1=s[::-1]

if(s==s1):

    print('Yes')

else:

    print('No')

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

**Ques:135(251)**

Given a string S of length N, print all permutations of the string in a separate line.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

123

OUTPUT

123

231

321

213

312

132

Test Cases:

Input 1 :

zmC

Output 1 :

zmC

zCm

mzC

mCz

Czm

Cmz

Input 2 :

eE

Output 2 :

eE

Ee

Input 3 :

pW

Output 3 :

pW

Wp

Input 4 :

fp

Output 4 :

fp

pf

Input 5 :

gRt

Output 5 :

gRt

gtR

Rgt

Rtg

tgR

tRg

Input 6 :

RS

Output 6 :

RS

SR

Input 7 :

qM

Output 7 :

qM

Mq

Input 8 :

KoZ

Output 8 :

KoZ

KZo

oKZ

oZK

ZKo

ZoK

Input 9 :

ma

Output 9 :

ma

am

Input 10 :

CBQ

Output 10 :

CBQ

CQB

BCQ

BQC

QCB

QBC

Input 11 :

JW

Output 11 :

JW

WJ

Input 12 :

Yp

Output 12 :

Yp

pY

Solution:

import itertools

a=str(input())

p=itertools.permutations(a)

for i in list(p):

    print(''.join(i))

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

**Ques:136(266)**

Form a number system with only 3 and 4. Find the nth number of the number system Eg.) The numbers are: 3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343,3344, 3433, 3434, 3443, 3444.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

1

OUTPUT

3

Test Cases:

Input 1 :

782

Output 1 :

433334444

Input 2 :

127

Output 2 :

3333333

Input 3 :

215

Output 3 :

4344333

Input 4 :

477

Output 4 :

44344443

Input 5 :

487

Output 5 :

44434333

Input 6 :

139

Output 6 :

3334433

Input 7 :

935

Output 7 :

443434333

Input 8 :

749

Output 8 :

344434443

Input 9 :

356

Output 9 :

34433434

Input 10 :

88

Output 10 :

344334

Input 11 :

194

Output 11 :

4333344

Input 12 :

203

Output 12 :

4334433

Solution:

l=[3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343,3344, 3433, 3434, 3443, 3444]

n=int(input())

print(l[n-1])

def find(n):

   arr = [''] \* (n + 1)

   size = 1

   m = 1

   while (size <= n):

       i = 0

       while(i < m and (size + i) <= n):

           arr[size + i] = "3" + arr[size - m + i]

           i += 1

       i = 0

       while(i < m and (size + m + i) <= n):

           arr[size + m + i] = "4" + arr[size - m + i]

           i += 1

       m = m << 1

       size = size + m

   print(arr[n])

n=int(input())

find(n)

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

**Ques:137(244)**

Given a number N followed by N numbers. Out of these N numbers some of them are repeated. Write a program to find the first number which is repeated. If no numbers are repeated print 'unique'.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

7

1 2 3 2 3 4 3

OUTPUT

2

Test Cases:

Input 1 :

7

3 3 1 1 2 4 2

Output 1 :

3

Input 2 :

7

1 3 1 3 2 2 4

Output 2 :

1

Input 3 :

14

10 10 11 4 9 6 3 7 10 2 2 9 11 11

Output 3 :

10

Input 4 :

8

4 2 1 4 2 3 1 2

Output 4 :

4

Input 5 :

11

3 2 8 4 8 2 7 5 7 1 4

Output 5 :

2

Input 6 :

7

4 1 2 2 4 4 2

Output 6 :

4

Input 7 :

9

5 5 2 5 5 6 5 6 2

Output 7 :

5

Input 8 :

7

4 4 4 1 4 3 4

Output 8 :

4

Input 9 :

5

1 1 1 1 1

Output 9 :

1

Input 10 :

6

2 1 1 1 1 1

Output 10 :

1

Input 11 :

8

5 1 3 5 1 5 2 3

Output 11 :

5

Input 12 :

13

10 6 4 6 3 1 6 5 3 8 6 1 5

Output 12 :

6

Solution:

n=int(input())

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

for i in l:

    if(l.count(i)!=1):

        print(i)

        break

else:

    print('unique')

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

**Ques:138(228)**

Given a number N and an array of size N(with both positive and negative integers), print the product of the elements in the maximum product subarray.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

5

1 2 3 4 5

OUTPUT

120

Test Cases:

Input 1 :

5

7 3 -5 6 0

Output 1 :

21

Input 2 :

5

1 -8 -3 5 0

Output 2 :

120

Input 3 :

7

-4 -3 3 10 3 4 -3

Output 3 :

4320

Input 4 :

11

-5 -4 -9 -10 -3 -10 8 1 -8 -2 -6

Output 4 :

8294400

Input 5 :

10

7 -9 -6 -6 -8 -5 2 -2 -7 9

Output 5 :

362880

Input 6 :

5

7 -1 -2 9 -5

Output 6 :

126

Input 7 :

10

-9 6 -3 -7 8 5 -5 -2 -4 3

Output 7 :

5443200

Input 8 :

9

-9 1 -1 -5 -8 9 -3 4 -7

Output 8 :

272160

Input 9 :

5

8 3 4 7 -9

Output 9 :

672

Input 10 :

10

8 -4 0 -4 5 -4 0 -8 -8 -6

Output 10 :

80

Input 11 :

7

-9 5 -2 8 -5 -6 4

Output 11 :

86400

Input 12 :

13

2 -3 10 -6 8 -4 5 4 8 -9 -3 4 9

Output 12 :

298598400

Solution:

n=int(input())

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

x=l.count(0)

for i in range(x):

    l.remove(0)

l.sort()

c=0

p=1

p1=[]

for j in l:

    p=p\*j

p1.append(p)

for i in range(len(l)):

    p=1

    l1=l[0:i]+l[i+1:]

    for j in l1:

        p=p\*j

    p1.append(p)

print(max(p1))

def maxsubarrayproduct(arr):

   n = len(arr)

   max\_ending\_here = 1

   min\_ending\_here = 1

   max\_so\_far = 1

   flag = 0

   for i in range(0, n):

       if arr[i] > 0:

           max\_ending\_here = max\_ending\_here \* arr[i]

           min\_ending\_here = min (min\_ending\_here \* arr[i], 1)

           flag = 1

       elif arr[i] == 0:

           max\_ending\_here = 1

           min\_ending\_here = 1

       else:

           temp = max\_ending\_here

           max\_ending\_here = max (min\_ending\_here \* arr[i], 1)

           min\_ending\_here = temp \* arr[i]

       if (max\_so\_far < max\_ending\_here):

           max\_so\_far = max\_ending\_here

   if flag == 0 and max\_so\_far == 1:

       return 0

   return max\_so\_far

n=int(input())

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

print(maxsubarrayproduct(l))

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

Ques:139(291)

Email Validation :

Given a string S check if is a valid email id based on the following Conditions.

1)@ should be present;

2)@ & . should not be repeated;

3)there should be atleast four characters between @ and .;

4)there should be at-least 3 characters before @ ;

5)the end of mail id should be .com;

If its a valid email id print 'yes' else print 'no'.

Input Size : |S| <= 100000

Sample Testcase :

INPUT

test@gmail.com

OUTPUT

yes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import re

regex = '^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$'

def check(email):

    if(re.search(regex,email)):

        print("yes")

    else:

        print("no")

email=input()

check(email)

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

Ques:140(229)

Given a number N and an array of N elements, a selection algorithm is implemented on this array where numbers at even position would be chosen, the algorithm is again and again implemented on the newly chosen array until only 1 element is remaining. Print the original position(index) of this element in the initial array.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

6

1 2 3 4 5 6

OUTPUT

3

Solution:

n=int(input())

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

d=l

while(len(d)>1):

    d=d[1::2]

for i in d:

    print(l.index(i))

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

Ques:141(323)

Given a number N print the right-angled triangle with the top level having N 1's followed by each level with is one 1 lesser.

Input Size : N <= 1000

Sample Testcase :

INPUT

3

OUTPUT

1 1 1

1 1

1

Solution:

r=int(input())

for i in range(r):

    for j in range(i-1,r-1):

        print('1',end='')

    print()

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

Ques:142(489)

Bala is the CEO of a company and decides to reward a few hardworking employees with an extra bonus.  He wants to find out the number of employees who are working parallely on two different projects and give them a bonus. The ids of employees working on a particular project are stored in an array.Given two arrays project1 and project2, find out whether the employees working on project2 are a subset of employees working on project1.

Constraints:

1 <= project1[i], project2[j] <= 100

Input Description:

First line denotes the number of employees working on project 1. Second line contains the employee ids of the employees working on project 1. Third line denotes the number of employees working on project 2 and 4th line contains employee ids of the employees working on project 2.

Output Description:

(yes/no) Whether the second array is a subset of the first array.

Sample Input :

3

1 2 3

2

1 2

Sample Output :

yes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

a=list(map(int,input().split()))

m=int(input())

b=list(map(int,input().split()))

A=set(a)

B=set(b)

if(B.issubset(A)):

    print('yes')

else:

    print('no')

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

Ques:143(320)

Alternate sorting:Given a number N followed by array of N elements,sort the array in such a way that the first number is the first maximum and second number is the 1st minimum 3rd number isthe 2nd maximum and so on.

Input Size : N <= 100000

Sample Testcase :

INPUT

8

7 623 19 10 11 9 3 15

OUTPUT

623 3 19 7 15 9 11 10

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

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

l.sort()

i=0

j=n-1

while(i<j):

    print(l[j],end=" ")

    j=j-1

    print(l[i],end=" ")

    i=i+1

if(n%2!=0):

    print(l[i])

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

Ques:144(280)

Given a number N followed by an unsorted array of N numbers and a number K, find if there exists a pair of elements in the array whose difference is K. Return count of such pairs.

Example k=4 and a[]={7,6,23,19,10,11,9,3,15}

Output should be : 6

Pairs can be:

7,11

7,3

6,10

19,23

15,19

15,11

Input Size : N <= 100000

Sample Testcase :

INPUT

6 4

8 12 16 4 0 20

OUTPUT

5

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

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

x=l.count(0)

for i in range(x):

    l.remove(0)

l.sort()

c=0

p=1

p1=[]

for j in l:

    p=p\*j

p1.append(p)

for i in range(len(l)):

    p=1

    l1=l[0:i]+l[i+1:]

    for j in l1:

        p=p\*j

    p1.append(p)

print(max(p1))

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

Ques:145(498)

Given a string containing a mix of upper and lowercase characters, sort the letters in ascending order such that the positions of upper and lowercase letters do not change. (eg) If the string is guVIONLineCoDIng, then the output must be  egCDIILginNnOVou

Input Description:

A string

Output Description:

Letters of the string sorted such that positions of upper and lower characters do not change.

Sample Input :

GuVi

Sample Output :

GiVu

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

def getSortedString(s, n):

    v1=[]

    v2=[]

    for i in range(n):

        if (s[i].islower()):

            v1.append(s[i])

        if (s[i].isupper()):

            v2.append(s[i])

    v1=sorted(v1)

    v2=sorted(v2)

    i = 0

     j = 0

    for k in range(n):

        if (s[k].islower()):

            s[k] = v1[i]

            print(s[k],end="")

            i+=1

        elif (s[k].isupper()):

            s[k] = v2[j]

            print(s[k],end="")

            j+=1

s = input()

ss=[i for i in s]

n = len(ss)

getSortedString(ss, n)

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

Ques:146(497)

Varsha is a Machine learning scientist. She wants to apply a few ML algorithms on the dataset to do some research and for that, she wants to merge two given arrays and sort them in ascending order. You are an intern working under Varsha and she has asked for your help for the same. Given 2 arrays arr1[] and arr2[], find the union of both the arrays sorted in ascending order.  Note: Union of two arrays must have distinct elements.

Input Description:

Size of arr1[] followed by elements of arr1. Size of arr2[] followed by elements of arr2.

Output Description:

Union of arr1 and arr2 sorted in ascending order.

Sample Input :

2

87 78

3

1 2 3

Sample Output :

1 2 3 78 87

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

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

n1=int(input())

l1=list(map(int,(input().split())))

l2=list(set().union(l,l1))

l2.sort()

print(\*l2)

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

Ques:147(487)

Jaspreet is a marketing manager and on inspecting the product ids of products documented as ‘sold’, he finds out that the sales person has repeated a few of the product ids. He decides to warn the sales person who has repeated the same product id thrice. Your task is to find out the number of triplets(ids occurring thrice) of product ids recorded.

Constraints:

1 <= N <= 100

1 <= A[] <= 1000

Input Description:

Number of product ids followed by the list of product ids

Output Description:

Number of triplets detected.

Sample Input :

10

2 1 1 3 1 8 7 2 2 6

Sample Output :

2

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

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

d=[]

for i in l:

    if(l.count(i)==3):

        if i not in d:

            d.append(i)

print(len(d))

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

Ques:148(590)

Rohan’s teacher wants to teach him about permuting the numbers.He is given an assignment,in which he is given a number,His task is to print the sum of all permutations that can be made up using those digits.

Input Description:

You are given a number

Output Description:

Your task is to print the sum of all permutations of number.

Sample Input :

18

Sample Output :

99

Solution:

import itertools

a=str(input())

p=itertools.permutations(a)

l=[]

for i in list(p):

    s=''.join(i)

    d=int(s)

    l.append(d)

print(sum(l))

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

Ques:149(502)

Vishal is a competitive coder and he loves problems on sorting and searching. He is bored right now and decides to solve a sorting problem with a modification. He decides to sort the elements at the even indices of an array in ascending order and the elements at the odd indices in descending order. Vishal goes outside his room after coding the solution, but his laptop crashes and he is unable to show it to his teacher. Can you help Vishal by coding the solution to the problem?

Constraints:

0<=array[i]<=10000

Input Description:

Size of the array followed by the elements of the array

Output Description:

Sorted elements of the array

Sample Input :

10

9 8 7 1 2 3 6 5 4 10

Sample Output :

2 10 4 8 6 5 7 3 9 1

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

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

a=[]

b=[]

for i in range(len(l)):

    if(i%2==0):

        a.append(l[i])

    else:

        b.append(l[i])

c=[]

a.sort()

b.sort(reverse=True)

j=0

for j in range(len(b)):

    c.append(a[j])

    c.append(b[j])

    j=j+1

if(len(a)>len(b)):

    c.append(a[-1])

print(\*c)

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

Ques:150(503)

You are given an array of N elements. Your task is to sort a subarray denoted by the index values i1 and i2 in descending order. Note: Index values start at 0. The sorting must be completed in O(i2-i1+1)log(i2-i1+1)

Input Description:

Size of array followed by elements of the array. The third line contains the start and end indices of the subarray( i1 and i2)

Output Description:

A partially sorted array

Sample Input :

6

34 9 2 10 56

1 3

Sample Output :

34 2 9 10 56

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import numpy as np

n=int(input())

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

i1,i2=map(int,(input().split()))

new=[]

a=np.array(l)

a[i1:i2+1].sort()

for i in a:

    new.append(i)

print(\*new)

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

Ques:151(504)

Given an array of N elements, sort the elements of the array based on frequency. If two numbers have the same frequency,then the smaller number comes first (eg) if the elements are 1,1,3,1,2,3,4 then the output is 2,4,3,3,1,1,1

Input Description:

Size of the array followed by the number of elements

Output Description:

Array elements sorted based on increasing frequency

Sample Input :

5

8 8 1 1 3

Sample Output :

3 1 1 8 8

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import collections

def frequency\_sort(a):

    f = collections.Counter(a)

    a.sort(key = lambda x:(f[x], x))

    return a

n=int(input())

a=list(map(int,input().split()))

print(\*frequency\_sort(a))

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

Ques:152(355)

Check whether the given 4 points form a square or not.

Example:

INPUT

10 10

10 20

20 20

20 10

OUTPUT

Yes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

l1=list(map(int,(input().split())))

l2=list(map(int,(input().split())))

l3=list(map(int,(input().split())))

l4=list(map(int,(input().split())))

if(l1[0]==l4[-1] and l1[-1]==l2[0] and l2[-1]==l3[0] and l3[-1]==l4[0]):

    print('yes')

else:

    print('no')

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

Ques:153(629)

You are given a string of different type of brackets. Your task is to check whether the given string is balanced or not balanced.

A string is said to be balanced if the number of opening brackets are equal to the number of closing brackets where the brackets should be of same kind.

Input Description:

You are given a string ‘s’.

Output Description:

Print 'yes' if the given string is balanced and no if it is not

Sample Input :

{}(())[][][{}]

Sample Output :

Yes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=input()

c=0

count=0

for i in n:

    if(i=='(' or i=='{' or i=='['):

        c=c+1

for j in n:

    if(j==')' or j=='}' or j==']'):

        count=count+1

if(c==count):

    print('yes')

else:

    print('no')

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

Ques:154(341)

Given 2 numbers N and K followed by N numbers such that sum of two pairs in the N numbers results K value, if it exist print 'yes' otherwise 'no'.

Input Size : N<=100000

Example:

INPUT

4 4

1 2 2 4

OUTPUT

Yes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n,k=map(int,(input().split()))

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

l2=[]

a=int(k)

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

    for j in range(i+1,len(l)):

        l1=l[i]+l[j]

        l2.append(l1)

for i in l2:

    if(i==a):

        print('yes')

             break

     else:

        print('no')

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

Ques:155(647)

You are given with list of numbers,Your task is to find the maximum count of the numbers which are consecutively increasing in list

Input Description:

You are given a number n denoting size of array,Next line contains ‘n’ space separated numbers.

Output Description:

Print the desired result

Sample Input :

6

1 2 0 3 4 5

Sample Output :

4

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

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

print(c)

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

Ques:156(583)

Jack is fond of numbers. He has given an array.He has to find the maximum length of sub array such that sum of elements in that sub array gives output 0.He is in little trouble, help him in finding solution

Input Description:

You are given an integer ‘n’ denoting the number of elements in array.Next line contains n space separated integers

Output Description:

Print the max length of that array with sum 0 and print -1 if not possible

Sample Input :

6

-4 3 1 0 0 6

Sample Output :

5

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

a=list(map(int,input().split()))

sub=[]

for i in range(len(a)+1):

    for j in range(i+1,len(a)+1):

        s=a[i:j]

        if(sum(s)==0):

            sub.append(len(s))

print(max(sub))

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

Ques:157(356)

Recently Kohli bought a very interesting book. He knows that it will take t seconds to read the book. Kohli wants to finish reading as fast as he can.But he has some work to do in each of n next days. The number of seconds that Kohli has to spend working during i-th day is ai. If some free time remains, he can spend it on reading. Help Kohli to determine the minimum number of day when He finishes reading. It is guaranteed that the answer doesn't exceed n. Remember that there are 86400 seconds in a day.

Input Size : N<=100T<=1000000

example

INPUT

2 2

86400 86398

OUTPUT

2

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

a,n=map(int,(input().split()))

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

m=sum(l)

i=0

day=86400

while(m>0):

    b=day-(a\*i)

    i+=1

    m=m-b

print(i)

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

Ques:158(356)

Recently Kohli bought a very interesting book. He knows that it will take t seconds to read the book. Kohli wants to finish reading as fast as he can.But he has some work to do in each of n next days. The number of seconds that Kohli has to spend working during i-th day is ai. If some free time remains, he can spend it on reading. Help Kohli to determine the minimum number of day when He finishes reading. It is guaranteed that the answer doesn't exceed n. Remember that there are 86400 seconds in a day.

Input Size : N<=100T<=1000000

example

INPUT

2 2

86400 86398

OUTPUT

2

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

a,n=map(int,(input().split()))

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

m=sum(l)

i=0

day=86400

while(m>0):

    b=day-(a\*i)

    i+=1

    m=m-b

print(i)

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

**TOPIC: STRINGS**

**Ques:159(657)**

**You are given a string ‘s’, find the minimum number of characters to be inserted to convert it to palindrome.**

**Input Description:**

**You are given a string ‘s’**

**Output Description:**

**Print 0 if no extra character is required else print the amount of characters required.**

**Sample Input :**

**abcd**

**Sample Output :**

**3**

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

def m(a,b):

    return min(a,b)

def Mininsertion(s,n):

    table=[[0 for i in range(n)]

              for i in range(n)]

    l,h,g=0,0,0

    for g in range(1,n):

        l=0

        for h in range(g,n):

            if(s[l]==s[h]):

                table[l][h]=table[l+1][h-1]

             else:

                table[l][h]=(m(table[l][h-1],table[l+1][h])+1)

            l=l+1

    return table[0][n-1];

s=input()

n=len(s)

print(Mininsertion(s,n))

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

Ques:160(684)

Given a string S consisting of lowercase latin letters, arrange all its letters in lexographical order using Counting Sort.

Input Description:

First Line contains positive integer N denoting the length of string.The second line of input contains the string S.

Output Description:

Print the sorted string.

Sample Input :

5

edsab

Sample Output :

abdes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n=int(input())

s=input()

l=sorted(s)

new=''

for i in l:

  new=new+i

print(new)

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

Ques:161(694)

You are given a string ‘s’. You have to tell whether the string can be arranged in last to first order.A last to first order is order in which if word 1 ends in say(char ‘a’) then the other word must start in char ‘a’.

Input Description:

You are given a string ‘s’.

Output Description:

Print ‘1’ if string can be arranged in last to first sequence and

Sample Input :

Loan avail

Sample Output :

1

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

s1,s2=map(str,(input().split()))

a=s1[0]

b=s2[-1]

c=s1[-1]

d=s2[0]

a1=a.lower()

b1=b.lower()

c1=c.lower()

d1=d.lower()

if(a1==b1 or c1==d1):

    print(1)

else:

    print(0)

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

Ques:162(289)

Given an arraylist A of string type which has name#mark1#mark2#mark3 format. Retrieve the name of the student who has scored max marks(total of three).

for eg: input: {'arun#12#12#12','deepak#13#12#12'}

output: Deepak

Input Size : A <= 100000

Sample Testcase :

INPUT

arun#12#12#12

deepak#13#12#12

OUTPUT

deepak

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import re

s1=input()

s2=input()

a1=re.findall(r'\d+',s1)

d1=list(map(int,a1))

n1=re.findall(r'\w+',s1)

name1=list(map(str,n1))

a2=re.findall(r'\d+',s2)

d2=list(map(int,a2))

n2=re.findall(r'\w+',s2)

name2=list(map(str,n2))

t1=sum(d1)

t2=sum(d2)

if(t1>t2):

    print(name1[0])

else:

    print(name2[0])

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

Ques:163(351)

Given a string S, check whether the given string is an anagram of the string “dhoni”. Anagram of a given string is a permutation of the set of characters present in the string. For example for the string “baba” aabb and baab are anagrams while aaab and bacd are not.

Input Size : |S|<=100000

Example:

INPUT

inohd

OUTPUT

Yes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

s1='dhoni'

a=list(s1)

a.sort()

s2=input()

b=list(s2)

b.sort()

if(a==b):

    print('yes')

else:

    print('no')

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

Ques:164(702)

You are given two numbers. Your task is to multiply the two numbers and print the answer.

Input Description:

You are given two numbers ‘n’ and ‘m’.

Output Description:

Print the multiplied answer

Sample Input :

99999 99998

Sample Output :

9999700002

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

n,m=map(int,(input().split()))

p=n\*m

print(p)

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

Ques:165(291)

Email Validation :

Given a string S check if is a valid email id based on the following Conditions.

1)@ should be present;

2)@ & . should not be repeated;

3)there should be atleast four characters between @ and .;

4)there should be at-least 3 characters before @ ;

5)the end of mail id should be .com;

If its a valid email id print 'yes' else print 'no'.

Input Size : |S| <= 100000

Sample Testcase :

INPUT

test@gmail.com

OUTPUT

Yes

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import re

regex = '^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$'

def check(email):

    if(re.search(regex,email)):

        print("yes")

    else:

        print("no")

email=input()

check(email)

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

Ques:166(297)

Given a string S of length N, reverse every word in place.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

abcd er x

OUTPUT

dcba re x

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

s=input()

a=s.split()

l=[]

for i in a:

    a=i[::-1]

    l.append(a)

print(\*l)

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

Ques:167(324)

Given 2 strings S and A find the longest common substring.If no common substring is found print '-1'

Input Size : |s| <= 1000, |A| <= 1000

Sample Testcase :

INPUT

codekata code

OUTPUT

Code

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

s1,s2=map(str,(input().split()))

l=""

for i in s1:

    if i in s2:

        l=l+i

if(len(l)==0):

    print(-1)

else:

    print(l)

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

Ques:168(250)

Given a string S of length N, find whether the given string is a palindrome using stack or linked list and print 'yes' otherwise print 'no'.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

GuviGeek

OUTPUT

No

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

s=input()

s1=s[::-1]

if(s==s1):

    print('yes')

else:

    print('no')

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

Ques:169(251)

Given a string S of length N, print all permutations of the string in a separate line.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

123

OUTPUT

123

231

321

213

312

132

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import itertools

a=str(input())

p=itertools.permutations(a)

for i in list(p):

    print(''.join(i))

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

Ques:170(316)

Given a string S, remove the characters which exist more than one times,and print the remaining string.

Input Size : |S| <= 10000000

Sample Testcase :

INPUT

Engineering

OUTPUT

Er

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import re

s=input()

a=re.findall(r'\w',s)

d=list(map(str,a))

l=[]

for i in d:

    if(d.count(i)==1):

        l.append(i)

r=''.join(l)

print(r,end="")

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

Ques:171(269)

Given a sequence of strings remove extra spaces in the string(there can be atmost one space be

Input Size : 1<= N <= 100000

Sample Testcases :

INPUT

aa abba abba

OUTPUT

aa abba abba

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import re

s=input()

a=re.sub(' +','',s)

print(a)

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

Ques:172(321)

Given a string S convert each characters of the string into ASCII values and print the sum of the numbers.

Input Size : |s| <= 100000

Sample Testcase :

INPUT

guvi

OUTPUT

443

Test Cases:

Input 1:

Output 1:

Input 2:

Output 2:

Input 3:

Output 3:

Input 4:

Output 4:

Input 5:

Output 5:

Input 6:

Output 6:

Input 7:

Output 7:

Input 8:

Output 8:

Input 9:

Output 9:

Input 10:

Output 10:

Solution:

import re

s=input()

a=re.findall(r'\w',s)

d=list(map(str,a))

sum=0

for i in d:

    b=ord(i)

    sum=sum+b

print(sum)

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

Ques:173(317)

Given a sentence S.check whether it is in camelcase.print 'yes' if it is in camelcase otherwise print 'no'.

input size : |s| <= 100000

Sample Testcase :

INPUT

CodekataChallenge

OUTPUT

Yes

Solution:

def camel(s):

    if (s != s.lower() and s != s.upper() and "\_" not in s):

        print('yes')

    else:

        print('no')

s=input()

camel(s)

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

Ques:174(322)

Given a string S consisting of 2 words reverse the order of two words .

Input Size : |S| <= 10000000

Sample Testcase :

INPUT

hello world

OUTPUT

world hello

Solution:

s=input()

a=s.split()

b=a[::-1]

print(\*b)

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

Ques:175(230)

Given a string S of length N, find the length of the largest alternating 'ab' subarray.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

ababababa

OUTPUT

9

Solution:

s=input()

c=0

for i in s:

    if(i=='a' and i=='b'):

        c=c+1

print(c)

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

Ques:176(270)

Given 2 strings S1 and S2. Find if String2 is substring of String1. If it is print the index of the first occurrence. else print -1.

Input Size : 1<= N <= 100000

Sample Testcases :

1)INPUT

test123string

123

OUTPUT

4

Solution:

s1=input()

s2=input()

a=s1.find(s2)

if(a!=-1):

    print(a)

else:

    print(-1)

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

Ques:177(293)

Given a string S of length N, reverse the words at odd positions.

Input Size: 1 <= N <= 100000

Sample Testcase :

INPUT

This is an example for this question

OUTPUT

sihT is na example rof this noitseuq

Solution:

s=input()

l=s.split()

new=[]

for i in range(len(l)):

    if(i%2==0):

        a=l[i]

        b=a[::-1]

        new.append(b)

    else:

        new.append(l[i])

print(\*new)

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

Ques:178(247)

Given a string S of length N, write a program that would reverse every word in the string.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

Hello World

OUTPUT

olleH dlroW

Solution:

s=input()

a=s.split()

l=[]

for i in a:

    a=i[::-1]

    l.append(a)

print(\*l)

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

Ques:179(273)

Given two numbers S1 and S2 represented as strings, return multiplication of the numbers as a string.

NOTE: The numbers can be arbitrarily large and are nonnegative.

Sample Testcase :

INPUT

2 10

OUTPUT

20

Solution:

s1,s2=map(str,(input().split()))

p=int(s1)\*int(s2)

print(p)

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

Ques:180(190)

Given two strings S1 and S2,display 'yes' if given two strings are complementary otherwise display 'no'. If we join alphabets of both the strings we should get all 26 capital letters exactly once, then only we can call them as complementary.

Sample Testcase :

INPUT

ABDCFGIJKLMNOPQUVWXYZ

EHRST

OUTPUT

Yes

Solution:

s1=input()

s2=input()

s3=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']

s4=s1+s2

S=list(s4)

S.sort()

if(S==s3):

    print('yes')

else:

    print('no')

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

Ques:181(225)

Given a string 'S', sort the characters based on the frequency(highest and lowest) and print the resultant string.(Note:If the frequency of different character is same then sort based on alphabetical order).

Input Size : 1 <= S <= 100000

Sample Testcases :

INPUT

aabbba

OUTPUT

Aaabbb

Solution:

import re

s=input()

a=re.findall(r'\w',s)

l=list(map(str,a))

l.sort()

b=''.join(l)

print(b,end="")

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

Ques:182(238)

Given a number N followed by N numbers. Find the largest number which can be formed from the given numbers and print it.

NOTE: Each number should be used exactly once.

Input Size : 1 <= N <= 100000

Sample Testcases :

INPUT

5

5 6 7 8 9

OUTPUT

98765

Solution:

n=int(input())

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

l.sort(reverse=True)

print(\*l,sep="")

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

Ques:183(186)

Given 2 numbers A,B. Print the GCD of A! and B!.

Sample Testcase :

INPUT

4 2

OUTPUT

2

Solution:

import math

a,b=map(int,(input().split()))

A=math.factorial(a)

B=math.factorial(b)

s=math.gcd(A,B)

print(int(s))

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

Ques:184(692)

You are given a string ‘s’.Determine whether the string given is made up of either

A

AB

ABB

If the string given is made up of any frequency occurrence of the above Print 1 else 0

Input Description:

You are given a string ‘s’.

Output Description:

Print 1 else 0

Sample Input :

AAAAA

Sample Output :

1

Solution:

s=input()

for i in s:

    if(i=='A' or i=='AB' or i=='ABB'):

        print('1')

        break

else:

    print('0')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ques:185(210)

Given a string S ,print the vowels first and then consonants in the same order as they have occurred in the string.

Input Size : N <= 10000

Sample Testcase :

INPUT

GuVI

OUTPUT

uIGV

Solution:

n=input()

v=['a','e','i','o','u','A','E','I','O','U']

l=""

l1=""

for i in n:

    if i in v:

        l=l+i

    else:

        l1=l1+i

print(l,end="")

print(l1)

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

Ques:186(655)

you are given with two strings.Your task is to check whether one of the string is substring of the other.If substring exists,then print the starting index of sub-string.

Input Description:

You are given two strings

Output Description:

Print the index of string if substring exists else print -1

Sample Input :

Yuvi

Yuvirat

Sample Output :

1

Solution:

s1=input()

s2=input()

a=s1.find(s2)

if(a!=-1):

    print(a+1)

else:

    print(-1)

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

Ques:187(673)

You are given a string ‘s’. Without spaces. Your task is to print all possible combination of strings by adding extra space.

Input Description:

You are given a string ‘s’.

Output Description:

Print the strings as described

Sample Input :

ABCD

Sample Output :

ABCD

ABC D

AB CD

AB C D

A BCD

A BC D

A B CD

A B C D

Solution:

from math import pow

def new(s):

    n=len(s)

    z=int(pow(2,n-1))

    for c in range(z):

        for j in range(n):

            print(s[j],end="")

            if(c&(1<<j)):

                print(" ",end="")

        print("\n",end="")

s=input()

new(s)

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

Ques:188(628)

 Joseph was going through topic of strings. He learnt about anagrams. But due to some circumstances he forget ,now he hired you to help him in completing the work.Your task is to tell whether the two given strings are anagrams

Input Description:

The first line of the input is a string N, the second line of the input is a string M

Output Description:

Compare the two string input N and M. Print 1 if they are anagram else print 0.

Sample Input :

abcd

cdab

Sample Output :

1

Solution:

a=str(input())

b=str(input())

A=list(a)

A.sort()

B=list(b)

B.sort()

if(A==B):

    print(1)

else:

    print(0)

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

Ques:189(138)

Given a number N and an array of N elements, print the suffix sum of the array.

Input Size : N <= 100000

Sample Testcase :

INPUT

4

2 4 4 2

OUTPUT

12 10 6 2

Solution:

n=int(input())

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

l1=[]

for i in range(n):

    l2=l[i:]

    l1.append(sum(l2))

print(\*l1)

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

Ques:190(159)

Given 2 strings S1 and S2,work on the strings such that both string has the same number of characters.To adjust the length reduce number of exceeding characters from longer string.

Sample Testcase :

INPUT

guvi geeks

OUTPUT

Guvigeek

Solution:

s1,s2=map(str,input().split())

l1=len(s1)

l2=len(s2)

if(l1==l2):

    print(s1,end="")

    print(s2)

if(l1>l2):

    l=l1-l2

    print(s1[:-l],end="")

    print(s2)

if(l2>l1):

    L=l2-l1

    print(s1,end="")

    print(s2[:-L])

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

Ques:191(149)

Given a date(DD-MM-YYYY),print the month in words.

Sample Testcase :

INPUT

01-01-2018

OUTPUT

January

Solution:

l=['January','Feburary','March','April','May','June','July','August','September','October','November','December']

d,m,y=map(int,(input().split('-')))

print(l[m-1])

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

Ques:192(203)

Let P represent Paper, R represent Rock and S represent Scissors. Given 2 out of the 3 determine which one wins. If its a draw print 'D'.

Sample Testcase :

INPUT

R P

OUTPUT

P

Solution:

a,b=map(str,(input().split()))

if((a=='R' and b=='R') or (a=='P' and b=='P') or (a=='S' and b=='S')):

    print('D')

elif(a=='R' and b=='P'):

    print('P')

elif(a=='R' and b=='S'):

    print('R')

elif(a=='S' and b=='P'):

    print('S')

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

Ques:193(193)

Roman Reigns want to identify the repeated letters in two given strings and capitalize it.Help him to achieve it.

Sample Testcase :

INPUT

computer program

OUTPUT

cOMPuteR PROgRaM

Solution:

s1,s2=map(str,input().split())

s3=s1+" "+s2

s4=[]

for i in s3:

    if(s3.count(i)>1):

        a=i.capitalize()

        s4.append(a)

    if(s3.count(i)==1):

        s4.append(i)

l=''.join(s4)

print(l)

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

Ques:194(124)

Given a number N, print the kth digit from the given position p(given order N P K).

Sample Testcase :

INPUT

5765 2 1

OUTPUT

6

Solution:

n,p,k=input().split()

l=list(n)

K=int(k)

P=int(p)

a=l[P:]

print(a[K-1])

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

Ques:195(167)

Given a string S, check whether it contains only the characters 'a' and 'b' or only 'a' or only 'b'

Input Size : |S| <= 100000

Sample Testcase :

INPUT

aabbaaab

OUTPUT

Yes

Solution:

import re

s=input()

a=re.findall(r'\w',s)

d=list(map(str,a))

D=len(d)

l=[]

for i in d:

    if(i=='a'):

        l.append(i)

    elif(i=='b'):

        l.append(i)

L=len(l)

if(L==D):

    print('yes')

else:

    print('no')

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

Ques:196(125)

Given a number N, print the sum of its first and last digit.

Input Size : |N| <= 10000

Sample Testcase :

INPUT

51233

OUTPUT

8

Solution:

import re

s=input()

a=re.findall(r'\d',s)

d=list(map(int,a))

sum=d[0]+d[-1]

print(sum)

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

Ques:197(87)

Given a string S change upper case to lowercase and lowercase to uppercase.

Input Size : |s| <= 10000000(complexity O(n))

Sample Testcase :

INPUT

CodEkaTa

OUTPUT

cODeKAtA

Solution:

s=input()

print(s.swapcase())

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

Ques:198(183)

Given a number N and an array of N strings,Print yes, if all strings have atleast one vowel in them otherwise print no.

Input Size : N <= 1000

Sample Testcase :

INPUT

5

code

overload

vishal

sundar

anish

OUTPUT

Yes

Solution:

n=int(input())

v=['a','e','i','o','u']

l=[]

for i in range(n):

    s=input()

    for j in s:

        if j in v:

            l.append(s)

            break

if(len(l)==n):

    print('yes')

else:

    print('no')

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

Ques:199(146)

Find the word having maximum length in a given sentence and print it. If two words are of same length return the first occuring word of max-length.

Input Size : |s| <= 100000

Sample Testcase :

INPUT

guvi geek

OUTPUT

Guvi

Solution:

s=input()

a=s.split()

l=list(a)

print(min(l))

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

Ques:200(142)

Accept a string and find if it is of date format 'dd/mm/yyyy'.

Sample Testcase :

INPUT

01/13/1999

OUTPUT

Solution:

d,m,y=map(int,(input().split('/')))

c=0

if(d<=31):

    c=c+1

if(m<=12):

    c=c+1

if(y>0):

    c=c+1

if(c==3):

    print('yes')

else:

    print('no')

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

Ques:201(68)

Given a string/sentence print its corresponding camelcase convention.

Input Size : |s| <= 1000000(complexity O(n))

Sample Testcase :

INPUT

guvi geeks

OUTPUT

GuviGeeks

Solution:

s=input()

a=s.capitalize()

print("".join(a))

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

Ques:202(189)

A number is given as input.Find the maximum number that can be formed using the digits.

Input Size : N <= 10000000

Sample Testcase :

INPUT

4123

OUTPUT

4321

Solution:

n=input()

l=list(map(int,str(n)))

l.sort(reverse=True)

print(\*l,sep="")

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

Ques:203(127)

Given a number N and a number K, check if it has all digits from 0 to k in it.

Input Size : N <= 100000

Sample Testcase :

INPUT

1234034 4

OUTPUT

Yes

Solution:

n,k=map(int,input().split())

l=[]

for i in range(k+1):

    l.append(i)

a=list(map(int,str(n)))

l1=[]

for j in a:

    if j in l:

        if j not in l1:

            l1.append(j)

if(len(l)==len(l1)):

     print('yes')

else:

    print('no')

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

Ques:204(128)

Given 2 strings S and X print the word position of X in S.(word count starts from 1).If the given word doesn't exists in S print '-1'.

Input Size : 1 <= |s|, |x| <= 1000

Sample Testcase :

INPUT

codekata coding challenge

coding

OUTPUT

2

Solution:

s=input()

x=input()

S=s.split()

X=x.split()

for i in S:

    if i in X:

        print(S.index(i)+1)

        break

else:

    print(-1)

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

Ques:205(137)

Given a number N and an array of N elements, print the prefix sum array.

Input Size : N <= 100000

Sample Testcase :

INPUT

4

2 4 4 2

OUTPUT

2 6 10 12

Solution:

n=int(input())

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

l1=[]

for i in range(n):

    l2=l[:i+1]

    l1.append(sum(l2))

print(\*l1)

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

Ques:206(129)

Given a binary number convert it to hexadecimal.

Sample Testcase :

INPUT

1100100

OUTPUT

64

Solution:

s=input()

b=int(s,2)

h=hex(b)

hexa=h[2:]

print(hexa)

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

Ques:207(75)

Given a string S,count the maximum number of times a character repeated in the string.If no character is repeated print '0'.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codekata

OUTPUT

2

Solution:

s=input()

l=[]

for i in s:

    a=s.count(i)

    l.append(a)

d=max(l)

if(d==1):

    print('0')

else:

    print(d)

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

Ques:208(70)

Given 2 strings check whether they differ exacly by one character.If yes then print 'yes' otherwise print 'no'

Input Size : |s| <= 100000(complexity O(nlogn) or O(n))

Sample Testcase :

INPUT

codekata codekate

OUTPUT

Yes

Solution:

s1,s2=map(str,input().split())

l1=list(s1)

l2=list(s2)

l3=[]

for i in l1:

    if i not in l2:

        l3.append(i)

if(len(l3)==1):

    print('yes')

else:

    print('no')

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

Ques:209(91)

Given a string S consisting of only '(' and ')', print 'yes' if it is balanced otherwise print 'no'.

Sample Testcase :

INPUT

(())

OUTPUT

Yes

Solution:

s=input()

count=0

c=0

for i in s:

    if(i=='('):

        count=count+1

        continue

for j in s:

    if(j==')'):

        c=c+1

if(count==c):

print('yes')

else:

print('no')

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

Ques:210(208)

Given a string S, retain the character(s) once irrespective of number of times it occurs in the given string.

Input Size : |S| <= 100000

Sample Testcase :

INPUT

aabbaa

OUTPUT

ab

Solution:

s=input()

a=list(s)

l=[]

for i in a:

    if i not in l:

        l.append(i)

s=''.join(l)

print(s,end="")

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

Ques:211(96)

Given a string, print the least repeated characters in the string.If there are more than one character repeated preserve the order as in the input.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codeKata challenge

OUTPUT

odKthng

Solution:

s=input()

A=s.replace(" ","")

l=list(A)

b=""

for i in l:

    if(l.count(i)==1):

        b=b+i

print(b)

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

Ques:212(90)

Given 2 strings and a number K, check whether they differ exactly by K characters.

Input Size : |s| <= 100000(complexity O(nlogn) or O(n))

Sample Testcase :

INPUT

codekata codeguvi 4

OUTPUT

Yes

Solution:

   s1,s2,k=[str(s1) for s1 in input().split()]

l1=list(s1)

l2=list(s2)

K=int(k)

l3=[]

for i in l1:

    if i not in l2:

        l3.append(i)

if(len(l3)==K):

    print('yes')

else:

    print('no')

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

Ques:213(195)

Given a string S, print 'yes' if the strings 'GUVI' and 'GEEK' is present case-sensitively in the string else print 'no'.

Input Size : 1 <= 100

Sample Testcase :

INPUT

Vishal\_Sundar prepared this question

OUTPUT

No

Solution:

import re

s=input()

a=re.findall(r'\w+',s)

d=list(map(str,a))

for i in d:

    if(i=='GUVI' and i=='GEEK'):

        print('no')

        break

else:

    print('yes')

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

Ques:214(38)

Given a string 'S' and a character 'K', find how many times 'K' got repeated in 'S'.If 'K' is not found in 'S' print -1

Input Size : |s| <= 100000

Sample Testcase :

INPUT

codekata a

OUTPUT

2

Solution:

s,s1=input().split()

a=list(s)

for i in a:

    if(i==s1):

        c=a.count(i)

        print(c)

        break

else:

    print('-1')

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

Ques:215(674)

You are given with string of words,we have to arrange them in reverse saturated order.

Input Description:

You are given a string ‘s’.

Output Description:

Print the reverse saturated order

Sample Input :

I am kohli fan

Sample Output :

I ma ilhok naf

Solution:

s=input()

a=s.split()

l=[]

for i in a:

    b=i[::-1]

    l.append(b)

print(\*l)

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

Ques:216(168)

Given a number N and an array of N strings, find if two consecutive words are same.

Input Size : N <= 1000

Sample Testcase :

INPUT

5

code

overload

vishal

sundar

anish

OUTPUT

no

Solution:

import re

n=int(input())

l=[]

for i in range(n):

    s=input()

    l.append(s)

l1=str(l)

a=re.findall(r'\w+',l1)

d=list(map(str,a))

for i in d:

    if(d.count(i)==2):

        print('yes')

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

Ques:217(73)

Given a number N, print the sum of the squares of its digits.

Input Size : 1 <= N <= 1000000000000000000

Sample Testcase :

INPUT

19

OUTPUT

82

Solution:

import re

s=input()

a=re.findall(r'\d',s)

d=list(map(int,a))

l=[]

for i in d:

    n=i\*i

    l.append(n)

sum=0

for j in l:

    sum=sum+j

print(sum)

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

Ques:218(480)

GIVEN A SET OF NUMBERS OF LENGTH N. YOUR TASK IS TO FIND THE quad ret (x1,x2,x3,x4) which will result the sum in 1. If quad ret exists print yes else print no

Input Description:

First line contains no. of test cases(T). Next line contains a number N. Next line contains n space separated numbers

Output Description:

PRINT YES OR NO

Sample Input :

1

5

1 -1 2 -2 3

Sample Output :

YES

Solution:

n1=int(input())

n=int(input())

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

sub=[]

for i in range(len(l)+1):

    for j in range(i+1,len(l)+1):

        s=l[i:j]

        sub.append(sum(s))

for i in sub:

    if(i==1):

        print("YES")

        break

else:

    print("NO")

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

Ques:219(315)

Given a String S and a string P, find if P is a substring of S. Print 'yes' if it is a substring else 'no'.

Input Size : |s| <= 10000 |p| <= 1000.

Sample Testcase :

INPUT

sundar sun

OUTPUT

Yes

Solution:

s1,s2=input().split()

if(s1.find(s2)==-1):

    print('no')

else:

    print('yes')

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

Ques:220(123)

Given a number N,check whether it has repeating digits in it.print 'yes' if it has repeating digits otherwise print 'no'.

Sample Testcase :

INPUT

11234

OUTPUT

Yes

Solution:

import re

s=input()

a=re.findall(r'\d',s)

d=list(map(int,a))

for i in d:

    if(d.count(i)!=1):

        print('yes')

        break

else:

    print('no')

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

Ques:221(164)

Print the position of first 1 from right to left, in binary representation of an Integer.

Sample Testcase :

INPUT

18

OUTPUT

2

Solution:

s=int(input())

b=bin(s)

d=b[2:]

r=d[::-1]

print(r.index('1')+1)

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

Ques:222(154)

Given 2 strings S,X. Print the string after deleting X.If X not found print the same string.

Input Size : 1 <= |s|, |x| <= 1000

Sample Testcase :

INPUT

Happy Birthday

Happy

OUTPUT

Birthday

Solution:

import re

s=input()

x=input()

a=re.findall(r'\w+',s)

d=list(map(str,a))

for i in d:

    if(i==x):

        b=s.replace(i,"")

        b1=b.replace(" ","")

        print(b1)

        break

else:

    print(s)

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

Ques:223(184)

Given 2 numbers N,K and an array of N integers, find if the element K exists in the array.

Input Size : N <= 100000

Sample Testcase :

INPUT

5 2

1 2 3 4 5

OUTPUT

yes

HINT: Read about Binary Search

Solution:

n,k=map(int,input().split())

s=list(map(int,input().split()))

for i in s:

    if(i==k):

        print('yes')

        break

else:

    print('no')

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

Ques:224(178)

Given a string and a number K.Print every kth character from the beginning.

Sample Testcase :

INPUT

string 3

OUTPUT

r g

Solution:

s,n=input().split(maxsplit=1)

k=int(n)

l=list(s)

for i in(k,len(l)):

    f=l[(k-1)::k]

print(\*f)

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

Ques:225(179)

Given a string and a number K, change every kth character to uppercase from beginning in string.

Sample Testcase :

INPUT

string 2

OUTPUT

sTrInG

Solution:

s,n=input().split(maxsplit=1)

k=int(n)

l=list(s)

for i in(k,len(l)):

    f=l[(k-1)::k]

a=[]

for i in l:

    if i in f:

        a.append(i)

b=[]

for j in l:

    if j in a:

        d=str(j)

        d1=d.upper()

        b.append(d1)

    if j not in a:

        d2=str(j)

        b.append(d2)

e="".join(b)

print(e)

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

Ques:226(169)

Given 2 numbers N,K and an array of N strings, find if any K consecutive strings are same.

Input Size : K <= N <= 1000

Sample Testcase :

INPUT

5 3

code

overload

vishal

vishal

vishal

OUTPUT

Yes

Solution:

import re

n,k=map(int,(input().split()))

l=[]

for i in range(n):

    s=input()

    l.append(s)

l1=str(l)

a=re.findall(r'\w+',l1)

d=list(map(str,a))

for i in d:

    if(d.count(i)==k):

        print('yes')

        break

else:

    print('no')

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

Ques:227(74)

Given a string S, print the reverse of the string after removing the vowels.If the resulting string is empty print '-1'.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codekata

OUTPUT

tkdc

Solution:

s=input()

v=('a','e','i','o','u','A','E','I','O','U')

for i in s:

    if i in v:

        s=s.replace(i,"")

s1=s[::-1]

if(len(s1)==0):

    print('-1')

else:

    print(s1)

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

Ques:228(36)

Given 2 strings S1 and s2, check whether they are case senitively equal without using any predefined function(case sensitive).If they are not same print 'no'

Sample Testcase :

INPUT

guvi guvi

OUTPUT

Yes

Solution:

s1,s2=map(str,(input().split()))

if(s1[::]==s2[::]):

  print('yes')

else:

  print('no')

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

Ques:229(145)

Given a input string S, reverse the given string by appending each character of the string with '-'.

Input Size : |S| <= 100000

Sample Testcase :

INPUT

codekata

OUTPUT

a-t-a-k-e-d-o-c

Solution:

n=input()

s=n[::-1]

r='-'.join(s[i:i+1] for i in range(0,len(s),1))

print(r)

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

Ques:230(166)

Given a number N, find the number of ones in its binary representation.

Sample Testcase :

INPUT

276

OUTPUT

3

Solution:

n=int(input())

b=bin(n)

for i in b:

a=b.count('1')

print(a)

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

Ques:231(95)

Given a string S, print the 1st and 3rd character of the string (chracter index starts from 1).

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codekata

OUTPUT

cdGiven a string S, print the 1st and 3rd character of the string (chracter index starts from 1).

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codekata

OUTPUT

cd

Solution:

n=input()

r=n[0]

s=n[2]

print(r,end="")

print(s)

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

Ques:232(624)

Indian PAN card issuing authority have found some fake PAN cards. They have hired you so that you can validate PAN card for them. Your task is to develop a suitable algorithm which could check if pan is valid or not

1)Pan must have uppercase letters only.

2)It must be of 10 character only

3)From index 1 to 5 all must be letters(A-Z),last index must be letter

4)Rest all must be integer Starting from 1

Input Description:

You are given a input string which indicates the PAN number

Output Description:

Print 'pan' if it is valid PAN number, else print 'not pan'

Sample Input :

HXTPS2142R

Sample Output :

pan

Solution:

n=input()

l=len(n)

if(l==10):

    if(n[0:5].isupper()):

        if(n[5:8].isdigit()):

            if(n[9].isupper()):

                print('pan')

            else:

                print('not pan')

        else:

            print('not pan')

    else:

        print('not pan')

else:

    print('not pan')

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

Ques:233(160)

Given 2 numbers n and m, n pairs of numbers a and b are given. In each pair 'a' means a person can start moving from point 'a' in the x axis to point 'b'(he can visit any point in between). Find if he can visit point m.

Sample Testcase :

INPUT

3 5

0 2

2 4

3 5

OUTPUT

Yes

Solution:

n,k=map(int,(input().split()))

c=0

l=[]

for i in range(n):

    l.append(list(map(int,(input().split()))))

for j in range(len(l)):

    for m in range(l[j][0],l[j][1]+1):

        if(m==k):

            c=c+1

if(c==1):

print('yes')

else:

print('no')

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

Ques:234(41)

Given 2 strings,check whether they have any common characters.If found print 'yes' else print 'no'.

Input Size : |s| <= 100000(O(n))

Sample Testcase :

INPUT

guvi guvigeeks

OUTPUT

Yes

Solution:

a,b=input().split()

x=0

for i in a:

    if i in b:

        print('yes')

        x=1

        break

if(x==0):

    print('no')

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

Ques:235(705)

You are given a string ‘s’.Print all the duplicate characters of string.

Input Description:

String ‘s; is given

Output Description:

Print only duplicate character and -1 if no character is duplicate.

Sample Input :

abcddee

Sample Output :

d e

Solution:

import re

s=input()

a=re.findall(r'\w',s)

d=list(map(str,a))

b=""

for i in d:

    if(d.count(i)>1):

        b=b+i

l=list(b)

l1=[]

for i in l:

    if i not in l1:

        l1.append(i)

if(len(l1)==0):

    print('-1')

else:

    print(\*l1)

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

Ques:236(128)

Given a binary number convert it into octal format.

Sample Testcase :

INPUT

1100100

OUTPUT

144

Solution:

s=input()

b=int(s,2)

o=oct(b)

print(o[2:])

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

Ques:237(39)

Given a sentence and string S, find how many times S occurs in the given sentence.If S is not found in the sentence print -1

Input Size : |sentence| <= 1000000(complexity O(n)).

Sample Testcase :

INPUT

I enjoy doing codekata

codekata

OUTPUT

1

Solution:

l=input()

s=input()

a=l.split()

for i in a:

    if i in s:

        b=a.count(i)

        print(b)

        break

else:

    print('-1')

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

Ques:238(672)

Given a number ‘n’, your task is to  generate all n-bit grey code sequences, “a grey code sequence is a sequence such that successive patterns in it differ by one bit”

Input Description:

You are given an number ‘n’.

Output Description:

Print the grey code sequence

Sample Input :

2

Sample Output :

00 01 11 10

Solution:

def gray\_code(n):

    if n <= 0:

        return []

    if n == 1:

        return ['0', '1']

    res = gray\_code(n-1)

    return ['0'+s for s in res] + ['1'+s for s in res[::-1]]

n=int(input())

print(\*gray\_code(n))

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

Ques:239(103)

Given 2 strings.check if the second string is a substring of the first string.Print 'yes' if there exists a valid substring otherwise print 'no'.

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

codekata code

OUTPUT

Yes

Solution:

s1,s2=input().split()

if(s1.find(s2)==-1):

    print('no')

else:

    print('yes')

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

Ques:240(88)

Given a string/sentence remove all the spaces and print the result.

Input Size : |s| <= 1000000(complexity O(n))

Sample Testcase :

INPUT

guvi geeks

OUTPUT

guvigeeks

Solution:

s=input()

print(s.replace(" ",""))

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

Ques:241(62)

Given a string S, print the reverse of the string.

Input Size : |s| <= 100000 (ie do it in O(n) or O(log n) time complexity)

Sample Testcase :

INPUT

codekata

OUTPUT

atakedoc

Solution:

s=input()

print(s[::-1])

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

Ques:242(86)

Given a sentence S take out the extra spaces.If no extra space is present print the same as output.

Input Size : |s| <= 100000(complexity O(n))

Sample Testcase :

INPUT

codekata challenge

OUTPUT

codekata challenge

Solution:

import re

s=input()

a=re.sub(' +', ' ', s)

print(a)

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

Ques:243(707)

You are given a string ‘s’. Your task is to print the characters which are not repeated.

Input Description:

You are given a string ‘s’.

Output Description:

Print the characters present once and -1 if there is no character which satisfy above condition

Sample Input :

dabbc

Sample Output :

d a c

Solution:

import re

s=input()

a=re.findall(r'\w',s)

d=list(map(str,a))

b=[]

for i in d:

    if(d.count(i)==1):

        b.append(i)

if(len(b)==0):

    print(-1)

else:

    print(\*b)

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

Ques:244(51)

Given a string two strings S1 and S2, remove characters from the S1 which are present in the S2.If S1 becomes empty then print -1

Input Size : N <= 100000

Sample Testcase :

INPUT

GUVI GEEK

OUTPUT

UVI

Solution:

s1,s2=input().split()

a=list(s1)

b=list(s2)

l=[]

for i in a:

    if i not in b:

        l.append(i)

if(len(l)>0):

    s=''.join(l)

    print(s.replace(" ",""))

else:

    print(-1)

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

Ques:245(78)

Given a number N and an array of N strings, find the number of strings that are an anagram of 'kabali'.If there exists no anagram for the given string print '0'.

Input Size : 1 <= N <= 1000

Sample Testcase :

INPUT

5

kabali

kaabli

kababa

kab

kabail

OUTPUT

3

Solution:

n=int(input())

l=[]

for i in range(n):

    a=input()

    s=sorted(a)

    l.append(s)

for j in l:

    a=l.count(j)

    if(a>1):

        print(a)

        break

else:

    print(0)

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

Ques:246(84)

Given a string S.Validate if a given string is numeric.print 'yes' if it is a numeric otherwise print 'no'.

Sample Testcase :

INPUT

guvigeeks

OUTPUT

No

Solution:

import re

s=input()

a=re.findall(r'\d',s)

d=list(map(int,a))

l=len(d)

if(l==0):

    print('no')

else:

    print('yes')

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

Ques:247(80)

Given a string S, print the encoded string by adding 3 to each character(a maps to d,b maps to e,c maps to f and so on).

Input Size : 1 <= N <= 100000

Sample Testcase :

INPUT

RADAR

OUTPUT

UDGDU

Solution:

s=input()

n=""

k=3

for i in range(len(s)):

    val=ord(s[i])

    dup=3

    if(val+3>122):

        k-=(122-val)

        k=k%26

        n+=chr(96+k)

    else:

         n+=chr(val+k)

    k=dup

print(n)

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

Ques:248(17)

Given a string S, print 2 strings such that first string containing all characters in odd position(s) and other containing all characters in even position(s).

Sample Testcase :

INPUT

XCODE

OUTPUT

XOE CD

Solution:

s=input()

l1=s[::2]

l2=s[1::2]

print(l1,end=" ")

print(l2)

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

Ques:249(708)

You are given two string ‘s1’ and ‘s2’. You have to tell whether these form pair of (strset) A pair of strings is said to be str set if one string is substring of other.

Input Description:

You are given two strings ‘s1’ and ‘s2’

Output Description:

Print Yes if they form strset and No if they don’t.

Sample Input :

abc ab

Sample Output :

Yes

Solution:

s1,s2=input().split()

if(s1.find(s2)==-1):

    print('No')

else:

    print('Yes')

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

Ques:250(71)

Given a day, print 'yes' if it is a holiday otherwise print'no'.Assume that weekend days are holidays

Sample Testcase :

INPUT

saturday

OUTPUT

yes

INPUT

monday

OUTPUT

No

Solution:

s=input()

if(s=='monday' or s=='tuesday' or s=='wednesday' or s=='thursday' or s=='friday'):

print('no')

elif(s=='saturday' or s=='sunday'):

print('yes')

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

Ques:251(696)

You are given a string ‘S’ consisting of lowercase Latin Letters. Find the first non repeating character in S. If you find all the characters are repeating print the answer as -1

Input Description:

You are given a string ‘s’

Output Description:

Print the first non occurring character if possible else -1.

Sample Input :

apple

Sample Output :

a

Solution:

s=input()

l=list(s)

a=[]

for i in l:

    if(l.count(i)==1):

        a.append(i)

        break

print(\*a)

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

Ques:252(67)

Given a string 'S' swap the even and odd characters starting from index 1(Assume the index starts from 0).

Input Size : |s| <= 10000000(complexity O(n))

Sample Testcase :

INPUT

codekata

OUTPUT

ocedakat

Solution:

s=input()

l=list(s)

for i in range(0,len(s),2):

    temp=l[i]

    l[i]=l[i+1]

    l[i+1]=temp

    s="".join(l)

print(s)

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

Ques:253(9)

Given a string S, print it after changing the middle element to \* (if the length of the string is even, change the 2 middle elements to \*).

Sample Testcase :

INPUT

hello

OUTPUT

he\*lo

Solution:

a=input()

s=list(a)

l=len(s)

if(l%2==0):

l=l//2

s[l]="\*"

s[l-1]="\*"

else:

l=int(l/2)

s[l]="\*"

print(''.join(s),end="")

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

Ques:254(652)

You are given a ‘true’ string. String is called true if weight of string is multiple of 8. Your task is to tell whether a string can be declared True or Not. Weight of string is the sum of ASCII value of Vowel character(s) present in the string.

Input Description:

You are given as string ‘s’ in lower cases

Output Description:

Print 1 for true and 0 for false

Sample Input :

raja

Sample Output :

0

Solution:

s=input()

a=list(s)

v=['a','e','i','o','u','A','E','I','O','U']

sum=0

for i in a:

    if i in v:

        b=ord(i)

        sum=sum+b

if(sum%8==0):

    print(1)

else:

    print(0)

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

Ques:255(704)

You are given two strings . Your task is to tell whether the pair of strings is panagram.

A pair of strings are said to be panagram if they both are palindrome and are anagram of each other.

Input Description:

You will be given two strings ‘s1’ and ‘s2’

Output Description:

Print 1 if they are panagram and 0 if they are not

Sample Input :

nitin intni

Sample Output :

1

Solution:

a,b=input().split()

if(len(a)!=len(b)):

    print(0)

s=sorted(a)

s1=sorted(b)

for i in range(0,len(a)):

    if(s[i]!=s1[i]):

        print(0)

print(1)

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

Ques:256(676)

You are given some words all in lower case letters your task is to print them in sorted order.

Input Description:

You are given a string ‘s’

Output Description:

Print the string in sorted order

Sample Input :

virat kohli

Sample Output :

kohli virat

Solution:

s=input()

a=s.split()

b=a[::-1]

print(\*b)

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

Ques:257(5)

Given a string S, print 'yes' if it is a palindrome or 'no' if it is not a palindrome.

Sample Testcase :

INPUT

lappal

OUTPUT

Yes

Solution:

s=input()

if(s[::1]==s[::-1]):

print('yes')

else:

print('no')

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

Ques:258(698)

You are given a string ‘s’.Your task is to find whether string is beautiful or not. A string is said to be beautiful whenever string is made up of only three characters. All the three characters must be distinct.Print true if string is beautiful and false when it is not beautiful

Input Description:

You are given a string

Output Description:

Print ‘1’ when string is beautiful and ‘0’ when it is not

Sample Input :

Aab

Sample Output :

1

Solution:

s=input()

a=list(s)

l=len(a)

if(l==3):

    if(a[0]!=a[1] and a[1]!=a[2] and a[2]!=a[0]):

        print('1')

    else:

        print('0')

else:

    print('0')

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

Ques:259(662)

You are given a paragraph.Your task is to print the words that come just after articles.

Input Description:

You are given a string ‘s’

Output Description:

print the words that come just after articles and -1 if there are no articles

Sample Input :

The sun rises in the east

Sample Output :

Sun east

Solution:

L=input().split()

L2=[]

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

    if L[i].lower() in ['a','an','the']:

        L2.append(L[i+1])

print(\*L2,end='')

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

Ques:260(664)

 You are given a string ‘s’.Your task is to print the string in alternate lowercase and uppercase order.

Input Description:

You are given a string

Output Description:

Print the string according to given criteria

Sample Input :

abcd efgh ijkl

Sample Output :

ABCD efgh IJKL

Solution:

s=input()

l=s.split()

b=[]

for i in range(len(l)):

    if(i%2!=0):

        b.append(l[i].lower())

    else:

        b.append(l[i].upper())

print(\*b)

s=input()

l=s.split()

b=[]

for i in range(len(l)):

    if(i%2!=0):

        b.append(l[i].lower())

    else:

        b.append(l[i].upper())

print(\*b)

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

Ques:261(475)

You are given with a string which comprises of some numbers. Your task is to find the largest integer present in the string.

Input Description:

First line contains n denoting number of Test Cases. The first and only Line of testcase has the string

Output Description:

Print the largest number

Sample Input :

I was born on 12 october 1998.

Sample Output :

1998

Solution:

import re

s=input()

a=re.findall(r'\d+',s)

d=list(map(int,a))

print(max(d))

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

Ques:262(508)

Ria is a 5 year old girl. Her mother wants to teach her how to sort words in the same order that they appear in a dictionary. She decides to write a program to sort a given set of strings based on their alphabetical order. Help Ria’s mother to complete the program.

Input Description:

A set of N strings

Output Description:

Alphabetically sorted set of strings

Sample Input :

3

InfinityWar EndGame Avengers

Sample Output :

Avengers EndGame InfinityWar

Solution:

import re

n=int(input())

l=input()

a=re.findall(r'\w+',l)

d=list(map(str,a))

d.sort()

print(\*d)

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

Ques:263(703)

You are given a string ‘s’.Your task is to print the string in the order they are present and then sum of digits.

Input Description:

You are given a string ‘s’.

Output Description:

Print the string and then at last sum of all the digits

Sample Input :

AC30BD40

Sample Output :

ACBD7

Solution:

l=list(input())

l1=[]

l2=[]

for i in l:

    if(i.isdigit()):

        l1.append(int(i))

    else:

        l2.append(i)

l2.append(str(sum(l1)))

s=''.join(l2)

print(s,end="")

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

Ques:264(644)

You are given a number with duplicate digits your task is to remove the immediate duplicate digits and print the result

Input Description:

You are given a long string of digits

Output Description:

Print the desired result print -1 if result length is 0

Sample Input :

1331

Sample Output :

-1

Solution:

import re

s=input()

b=""

temp = re.findall(r'\d',s)

d= list(map(str, temp))

new=""

for i in d:

    if(d.count(i)==1):

        new=new+i

if(len(new)==0):

    print('-1')

else:

   print(new)

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

Ques:265(670)

You are given a string ‘s’. Your task is to tell whether string is beautiful or not.A beautiful string is a string in which String starts with ‘a’ or ‘A’ and middle element is either ‘m’ or ‘M’ and last element is ‘z’or ‘Z’

Input Description:

You are given a string ‘s’.

Output Description:

Print 1 if string is beautiful and 0 if it is not

Sample Input :

Amz

Sample Output :

1

Solution:

s=input()

l=(len(s))//2

if(s.startswith('a') or s.startswith('A')):

    if(s.endswith('z') or s.endswith('Z')):

        if(s[l]=='m' or s[l]=='M'):

            print('1')

        else:

            print('0')

    else:

        print('0')

else:

    print('0')

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

Ques:266(669)

Rahul is given a task to manipulate a string, He hired you as a developer your task is to delete all the consecutive repeating characters and print the result left.

Input Description:

You are given a string ‘s’

Output Description:

Print the remaining string

Sample Input :

mississipie

Sample Output :

mpie

Solution:

s=input()

l=list(s)

for i in l:

    if(l.count(i)==1):

        print(i,end="")

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

Ques:267(656)

you are given a string made up of parenthesis only.Your task is to check whether parenthesis are balanced or not.If they are balanced print 1 else print 0

Input Description:

You are given a string ‘s’

Output Description:

Print 1 for balanced and 0 for imbalanced

Sample Input :

{({})}

Sample Output :

1

Solution:

s=input()

count=0

c=0

for i in s:

    if(i=='{' or i=='[' or i=='('):

        count=count+1

        continue

for j in s:

    if(j=='}' or j==']' or j==')'):

        c=c+1

if(count==c):

    print('1')

else:

    print('0')

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

Ques:268(634)

Joyal was given a sentence. His task is to delete the two words that comes together and print the sentence so that the words in the output sentence have distinct words compared to their adjacent words. If no words are present in the output sentence print -1

Input Description:

You are given input string 'S'

Output Description:

Print the all the words that are left in the string 's' so that the words in the output sentence have distinct words compared to their adjacent words. Print -1 if no words are left

Sample Input :

I am john cena cena john

Sample Output :

I am

Solution:

n=input().split()

l=['@']

for i in n:

    if i==l[-1]:

        l.remove(i)

    else:

        l.append(i)

if(len(l)==1):

    print(-1)

else:

    print(\*l[1:])

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

Ques:269(665)

You are given a string.Your task is to print only the consonants present in the string without affecting the sentence spacings if present. If no consonants are present print -1

Input Description:

You are given a string ‘s’.

Output Description:

Print only consonants.

Sample Input :

I am shrey

Sample Output :

m shry

Solution:

s=input()

l=[]

v=['a','e','i','o','u','A','E','I','O','U']

for i in s:

    if i in v:

        s=s.replace(i,"")

    else:

        l.append(i)

a=''.join(l)

print(a)

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

Ques:270(6)

Given a string S, print 'yes' if it has a vowel in it else print 'no'.

Sample Testcase :

INPUT

codekata

OUTPUT

Yes

Solution:

s=input()

for i in s:

    if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u'):

        print('yes')

        break

else:

    print('no')

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

Ques:271(650)

Given a string 'S' print the sum of weight of the String. A weight of character is defined as the ASCII value of corresponding character.

Input Description:

You are given a string ‘s’

Output Description:

Print weight

Sample Input :

abc

Sample Output :

294

Solution:

s=input()

d=list(s)

sum=0

for i in d:

    b=ord(i)

    sum=sum+b

print(sum)

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

Ques:272(614)

Radha newly learnt about palindromic strings.A palindromic string is a string which is same when read from left to right and also from right to left.Help her in implementing the logic.

Input Description:

You are given a String ‘s’

Output Description:

Print 1 if String is palindrome or 0 if not

Sample Input :

NITIN

Sample Output :

1

Solution:

s=input( )

if(s[::]==s[::-1]):

    print('1')

else:

    print('0')

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

Ques:273(547)

In XYZ country there is rule that car’s engine no. depends upon car’ number plate. Engine no is sum of all the integers present on car’s Number plate.The issuing authority has hired you in order to provide engine no. to the cars.Your task is to develop an algorithm which takes input as in form of string(Number plate) and gives back

Engine number.

Input Description:

You are given a string ’s’

Output Description:

Print the number plate

Sample Input :

HR05-AA-2669

Sample Output :

28

Solution:

import re

s=input()

a=re.findall(r'\d',s)

l=list(map(int,a))

sum=0

for i in l:

sum=sum+i

print(sum)

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

Ques:274(648)

You are given a string. You have to print “Wonder” if the string is wonderful and -1 if it is not. A wonderful string is a string,which is made up of exactly 3 characters.

Input Description:

You are given a string

Output Description:

Print “Wonder” if it is wonderful and -1 if it is not

Sample Input :

aabbcc

Sample Output :

Wonder

Solution:

s=input()

b=""

for i in s:

    if i not in b:

        b=b+i

s=len(b)

if(s==3):

    print('Wonder')

else:

    print('-1')

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