

100 Problems in Codeforces.com

2020 fall, Compiled by Hongfei Yan

How to find the problems?

Visit <http://codeforces.com/>, click 'PROBLEMSET', then click green checkmark (order=BY_SOLVED_DESC). That is, click http://codeforces.com/problemset?order=BY_SOLVED_DESC.

What is Codeforces? What kind of a site/resource is it?

Codeforces is a project joining people interested in and taking part in programming contests. On one hand, Codeforces is a social network dedicated to programming and programming contests. On the other hand, it is a platform where contests are held regularly, the participant's skills are reflected by their rating and the former contests can be used to prepare.

如果想查看某个题目的其他人提交的代码，例如查看580C 可以访问

<http://codeforces.com/problemset/status/580/problem/C>

Basic Programming Exercises

#. title, algorithm, difficulty, link

4A. Watermelon

brute force/math, 800, <http://codeforces.com/problemset/problem/4/A>

One hot summer day Pete and his friend Billy decided to buy a watermelon. They chose the biggest and the ripest one, in their opinion. After that the watermelon was weighed, and the scales showed w kilos. They rushed home, dying of thirst, and decided to divide the berry, however they faced a hard problem.

Pete and Billy are great fans of even numbers, that's why they want to divide the watermelon in such a way that each of the two parts weighs even number of kilos, at the same time it is not obligatory that the parts are equal. The boys are extremely tired and want to start their meal as soon as possible, that's why you should help them and find out, if they can divide the watermelon in the way they want. For sure, each of them should get a part of positive weight.

Input

The first (and the only) input line contains integer number w ($1 \leq w \leq 100$) — the weight of the watermelon bought by the boys.

Output

Print YES, if the boys can divide the watermelon into two parts, each of them weighing even number of kilos; and NO in the opposite case.

将一个数分成两个偶数

```
w = int(input())
if 3<=w<=100 and (w-2) % 2 == 0:
    print('YES')
else:
    print('NO')
```

short code 1

```
i=int(input())
print(['YES','NO'][i%2 or i<3])
```

short code 2

```
print("YNEOS"[2**int(input())%24<9::2])
```

1A. Theatre Square

math, 1000, <http://codeforces.com/problemset/problem/1/A>

Theatre Square in the capital city of Berland has a rectangular shape with the size $n \times m$ meters. On the occasion of the city's anniversary, a decision was taken to pave the Square with square granite flagstones. Each flagstone is of the size $a \times a$.

What is the least number of flagstones needed to pave the Square? It's allowed to cover the surface larger than the Theatre Square, but the Square has to be covered. It's not allowed to break the flagstones. The sides of flagstones should be parallel to the sides of the Square.

Input

The input contains three positive integer numbers in the first line: n , m and a ($1 \leq n, m, a \leq 109$).

Output

Write the needed number of flagstones.

用边长为a的正方形瓷砖铺满 $m \times n$ 的广场，按长和宽算个数

```
import math
n, m, a = [int(x) for x in input().split()]
l = math.ceil(n/a)
w = math.ceil(m/a)
print(l*w)
```

short code

```
n,m,a=map(int,input().split())
print(-n//a*(-m//a))
```

不用math.ceil, $-5//4=-2$, 运算顺序相当于 $(-n//a)*(-m//a)$

58A. Chat room

greedy/strings, 1000, <http://codeforces.com/problemset/problem/58/A>

Vasya has recently learned to type and log on to the Internet. He immediately entered a chat room and decided to say hello to everybody. Vasya typed the word *s*. It is considered that Vasya managed to say hello if several letters can be deleted from the typed word so that it resulted in the word "hello". For example, if Vasya types the word "ahhellllloou", it will be considered that he said hello, and if he types "hlelo", it will be considered that Vasya got misunderstood and he didn't manage to say hello. Determine whether Vasya managed to say hello by the given word *s*.

Input

The first and only line contains the word *s*, which Vasya typed. This word consists of small Latin letters, its length is no less than 1 and no more than 100 letters.

Output

If Vasya managed to say hello, print "YES", otherwise print "NO".

Examples

input

```
ahhellllloou
```

output

```
YES
```

input

```
hlelo
```

output

```
NO
```

2020fall-cs101, 李受禧

```
s = input()
s = s.lower()

dp = [0]*5
data = 'hello'
cnt = 0

for c in s:
    if c == data[cnt]:
        dp[cnt] += 1
        cnt += 1

    if cnt == 5:
        break

if sum(dp) == 5:
    print('YES')
else:
```

```
print('NO')
```

```
import re
s = input()
r = re.search('h.*e.*l.*l.*o', s)
print(['YES', 'NO'][r==None])
```

71A. Way Too Long Words

strings, 1000, <http://codeforces.com/problemset/problem/71/A>

Sometimes some words like "*localization*" or "*internationalization*" are so long that writing them many times in one text is quite tiresome.

Let's consider a word *too long*, if its length is **strictly more** than 10 characters. All too long words should be replaced with a special abbreviation.

This abbreviation is made like this: we write down the first and the last letter of a word and between them we write the number of letters between the first and the last letters. That number is in decimal system and doesn't contain any leading zeroes.

Thus, "*localization*" will be spelt as "*l10n*", and "*internationalization*" will be spelt as "*i18n*".

You are suggested to automatize the process of changing the words with abbreviations. At that all too long words should be replaced by the abbreviation and the words that are not too long should not undergo any changes.

Input

The first line contains an integer n ($1 \leq n \leq 100$). Each of the following n lines contains one word. All the words consist of lowercase Latin letters and possess the lengths of from 1 to 100 characters.

Output

Print n lines. The i -th line should contain the result of replacing of the i -th word from the input data.

长度大于10 的单词缩写为首字母+中间字母数+尾字母

```
n = int(input())
for i in range(n):
    word = input()
    length = len(word)
    if length > 10:
        num = str(length-2)
        short = word[0]+num+word[-1]
        print(short)
    else:
        print(word)
```

选择性输出, 可以在 print里完成

```
for _ in range(int(input())):
    a = input()
    l = len(a)
    print(a if l<11 else a[0]+str(l-2)+a[l-1])
```

231A. Team

bruteforce/greedy, 800, <http://codeforces.com/problemset/problem/231/A>

One day three best friends Petya, Vasya and Tonya decided to form a team and take part in programming contests. Participants are usually offered several problems during programming contests. Long before the start the friends decided that they will implement a problem if at least two of them are sure about the solution. Otherwise, the friends won't write the problem's solution.

This contest offers n problems to the participants. For each problem we know, which friend is sure about the solution. Help the friends find the number of problems for which they will write a solution.

Input

The first input line contains a single integer n ($1 \leq n \leq 1000$) — the number of problems in the contest. Then n lines contain three integers each, each integer is either 0 or 1. If the first number in the line equals 1, then Petya is sure about the problem's solution, otherwise he isn't sure. The second number shows Vasya's view on the solution, the third number shows Tonya's view. The numbers on the lines are separated by spaces.

Output

Print a single integer — the number of problems the friends will implement on the contest.

三人小队，至少两个人会才能答出问题

```
n = int(input())
num = 0
for i in range(n):
    a, b, c = [int(x) for x in input().split()]
    if a + b + c > 1:
        num += 1
print(num)
```

short code

```
print(sum(input().count('1')>1 for x in range(int(input()))))
```

print()也很耗时间，如果要多次输出，可以在print()里套循环，减少调用次数

158A. Next Round

*special problem/implementation, 800, <http://codeforces.com/problemset/problem/158/A>

"Contestant who earns a score equal to or greater than the k -th place finisher's score will advance to the next round, as long as the contestant earns a positive score..." — an excerpt from contest rules.

A total of n participants took part in the contest ($n \geq k$), and you already know their scores. Calculate how many participants will advance to the next round.

Input

The first line of the input contains two integers n and k ($1 \leq k \leq n \leq 50$) separated by a single space.

The second line contains n space-separated integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 100$), where a_i is the score earned by the participant who got the i -th place. The given sequence is non-increasing (that is, for all i from 1 to $n - 1$ the following condition is fulfilled: $a_i \geq a_{i+1}$).

Output

Output the number of participants who advance to the next round.

统计不小于第k位选手得分的人数

```
n, k = map(int, input().split())
score = [int(x) for x in input().split()]
num = 0
k_score = score[k-1]
for i in range(n):
    s = score[i]
    if s >= k_score and s > 0:
        num += 1
print(num)
```

short code

```
i=lambda:map(int,input().split())
n,k=i()
a=list(i())
print(sum([x>=(a[k-1] or 1) for x in a]))
```

用or不能用 and

118A. String Task

implementation/strings, 1000, <http://codeforces.com/problemset/problem/118/A>

Petya started to attend programming lessons. On the first lesson his task was to write a simple program. The program was supposed to do the following: in the given string, consisting of uppercase and lowercase Latin letters, it:

- deletes all the vowels,
- inserts a character "." before each consonant,
- replaces all uppercase consonants with corresponding lowercase ones.

Vowels are letters "A", "O", "Y", "E", "U", "I", and the rest are consonants. The program's input is exactly one string, it should return the output as a single string, resulting after the program's processing the initial string.

Help Petya cope with this easy task.

Input

The first line represents input string of Petya's program. This string only consists of uppercase and lowercase Latin letters and its length is from 1 to 100, inclusive.

Output

Print the resulting string. It is guaranteed that this string is not empty.

去掉元音，并用.来连接剩余字母的小写

```
str = input()
word = str.lower()
output = []
vowel = ['a', 'e', 'i', 'o', 'u', 'y']
for char in word:
    if char not in vowel:
        output.append('.')
        output.append(char)
print(''.join(output))
```

short code

```
print(''.join('.') + l for l in input().lower() if l not in 'aeiouy'))
```

50A. Domino piling

greedy/math, 800, <http://codeforces.com/problemset/problem/50/A>

You are given a rectangular board of $M \times N$ squares. Also you are given an unlimited number of standard domino pieces of 2×1 squares. You are allowed to rotate the pieces. You are asked to place as many dominoes as possible on the board so as to meet the following conditions:

1. Each domino completely covers two squares.
2. No two dominoes overlap.
3. Each domino lies entirely inside the board. It is allowed to touch the edges of the board.

Find the maximum number of dominoes, which can be placed under these restrictions.

Input

In a single line you are given two integers M and N — board sizes in squares ($1 \leq M \leq N \leq 16$).

Output

Output one number — the maximal number of dominoes, which can be placed.

不同于 Theatre Square，多米诺骨牌可以横放或竖放，可以不铺满（其实最多空一格）可直接按面积相除取整

```
M, N = [int(x) for x in input().split()]
print(int(M*N/2))
```

short code

```
print(eval('*'.join(input().split()))//2)
```

eval会把字符串当成算式计算也可返回相应的 list,tuple等

282A. Bit++

implementation, 800, <http://codeforces.com/problemset/problem/282/A>

The classic programming language of Bitland is Bit++. This language is so peculiar and complicated.

The language is that peculiar as it has exactly one variable, called *x*. Also, there are two operations:

- Operation ++ increases the value of variable *x* by 1.
- Operation -- decreases the value of variable *x* by 1.

A statement in language Bit++ is a sequence, consisting of exactly one operation and one variable *x*. The statement is written without spaces, that is, it can only contain characters "+", "-", "X".

Executing a statement means applying the operation it contains.

A programme in Bit++ is a sequence of statements, each of them needs to be executed. Executing a programme means executing all the statements it contains.

You're given a programme in language Bit++. The initial value of *x* is 0. Execute the programme and find its final value (the value of the variable when this programme is executed).

Input

The first line contains a single integer *n* ($1 \leq n \leq 150$) — the number of statements in the programme.

Next *n* lines contain a statement each. Each statement contains exactly one operation (++ or --) and exactly one variable *x* (denoted as letter «X»). Thus, there are no empty statements. The operation and the variable can be written in any order.

Output

Print a single integer — the final value of *x*.

定义两种运算, ++表示 +1, , --表示 -1 x的值不断更新

```
n = int(input())
x = 0
for i in range(n):
    statement = input()
    if '++' in statement:
        x += 1
    else:
        x -= 1
print(x)
```

short code

```
f=input
print(sum('+' in f() or -1 for i in range(int(f()))))
```


112A. Petya and Strings

implementation/strings, 800, <http://codeforces.com/problemset/problem/112/A>

Little Petya loves presents. His mum bought him two strings of the same size for his birthday. The strings consist of uppercase and lowercase Latin letters. Now Petya wants to compare those two strings lexicographically. The letters' case does not matter, that is an uppercase letter is considered equivalent to the corresponding lowercase letter. Help Petya perform the comparison.

Input

Each of the first two lines contains a bought string. The strings' lengths range from 1 to 100 inclusive. It is guaranteed that the strings are of the same length and also consist of uppercase and lowercase Latin letters.

Output

If the first string is less than the second one, print "-1". If the second string is less than the first one, print "1". If the strings are equal, print "0". Note that the letters' case is not taken into consideration when the strings are compared.

Input

Each of the first two lines contains a bought string. The strings' lengths range from 1 to 100 inclusive. It is guaranteed that the strings are of the same length and also consist of uppercase and lowercase Latin letters.

Output

If the first string is less than the second one, print "-1". If the second string is less than the first one, print "1". If the strings are equal, print "0". Note that the letters' case is not taken into consideration when the strings are compared.

按字典方式排序，不考虑大小写。一个个字母比即可

```
s1 = input().lower()
s2 = input().lower()
num = 0
for i in range(len(s1)):
    if ord(s1[i]) < ord(s2[i]):
        print('-1')
        break
    elif ord(s1[i]) > ord(s2[i]):
        print('1')
        break
    else:
        num += 1
if num == len(s1):
    print('0')
```

可以直接比较字符串的大小，不用一个一个比较（比较字符串大小本来就是一个一个比较的，这样代码会比较简）（来自江雨翔）

```

line1 = input()
line2 = input()
if line1.lower() > line2.lower():
    print(1)
elif line1.lower() < line2.lower():
    print(-1)
else:
    print(0)

```

short code

```

i=input();a=i().lower();b=i().lower()
print((a>b)-(a<b))

```

()表判断，返回布尔值，加减运算 True是 1 False是 0

263A. Beautiful Matrix

implementation, 800, <http://codeforces.com/problemset/problem/263/A>

You've got a 5×5 matrix, consisting of 24 zeroes and a single number one. Let's index the matrix rows by numbers from 1 to 5 from top to bottom, let's index the matrix columns by numbers from 1 to 5 from left to right. In one move, you are allowed to apply one of the two following transformations to the matrix:

1. Swap two neighboring matrix rows, that is, rows with indexes i and $i + 1$ for some integer i ($1 \leq i < 5$).
2. Swap two neighboring matrix columns, that is, columns with indexes j and $j + 1$ for some integer j ($1 \leq j < 5$).

You think that a matrix looks *beautiful*, if the single number one of the matrix is located in its middle (in the cell that is on the intersection of the third row and the third column). Count the minimum number of moves needed to make the matrix beautiful.

Input

The input consists of five lines, each line contains five integers: the j -th integer in the i -th line of the input represents the element of the matrix that is located on the intersection of the i -th row and the j -th column. It is guaranteed that the matrix consists of 24 zeroes and a single number one.

Output

Print a single integer — the minimum number of moves needed to make the matrix beautiful.

```

for i in range(5):
    s = input().split()
    if "1" in s:
        print(abs(i-2)+abs(s.index("1")-2))
        break

```

周晋飞

```

matrix = [[int(x) for x in input().split()] for i in range(5)]

for i in range(5):
    if 1 in matrix[i]:
        j = matrix[i].index(1)
        print(abs(i-2)+abs(j-2))
        break

```

马玉娇, 代码容易看懂

```

a = input().split()
b = input().split()
c = input().split()
d = input().split()
e = input().split()
if '1' in a:
    print(abs(a.index('1')-2) +2)
if '1' in b:
    print(abs(b.index('1')-2) +1)
if '1' in c:
    print(abs(c.index('1')-2))
if '1' in d:
    print(abs(d.index('1')-2) +1)
if '1' in e:
    print(abs(e.index('1')-2) +2)

```

韩无极, 代码容易看懂

```

for i in range(5):
    lis = list(map(int,input().split()))
    if 1 in lis:
        r = i
        break
for j in range(5):
    if lis[j]==1:
        c = j
print(abs(c-2)+abs(r-2))

```

庞翔升, 代码不容易看懂

```

c = 0
for i in range(5):
    exec('a%s=list(input().split())'%i)
for i in range(5):
    for o in range(5):
        exec('c=(a%s[o])'%i)
        if c=='1':
            print(abs(i-2)+abs(o-2))

```

```

mx = [ [0]*5 for row in range(5) ]

for row in range(5):
    line = [int(j) for j in input().split()]
    if max(line)==1:
        for column in range(5):
            if line[column]==1:
                mx[row][column]=1
                break
    print(abs(row-2) + abs(column-2))
    break

```

short code

```

l = [2,1,0,1,2]
for i in l:
    s = input()
    if "1" in s:
        print(i + l[s.find("1")//2])

```

339A. Helpful Maths

greedy/implementation/sortings/strings, 800, <http://codeforces.com/problemset/problem/339/A>

Xenia the beginner mathematician is a third year student at elementary school. She is now learning the addition operation.

The teacher has written down the sum of multiple numbers. Pupils should calculate the sum. To make the calculation easier, the sum only contains numbers 1, 2 and 3. Still, that isn't enough for Xenia. She is only beginning to count, so she can calculate a sum only if the summands follow in non-decreasing order. For example, she can't calculate sum $1+3+2+1$ but she can calculate sums $1+1+2$ and $3+3$.

You've got the sum that was written on the board. Rearrange the summands and print the sum in such a way that Xenia can calculate the sum.

Input

The first line contains a non-empty string s — the sum Xenia needs to count. String s contains no spaces. It only contains digits and characters "+". Besides, string s is a correct sum of numbers 1, 2 and 3. String s is at most 100 characters long.

Output

Print the new sum that Xenia can count.

```

s = input()
fq = 4*[0] # the first is no use.

for i in range(0, len(s), 2):
    fq[int(s[i])] += 1

ns = ''
for i in range(1,4) :
    while fq[i]>0 :
        fq[i] -= 1
        ns += str(i)

print('+'.join(ns))

```

short code

```

s = [int(n) for n in input().split('+')]

s.sort()

print('+'.join(str(i) for i in s))

```

281A. Word Capitalization

implementation/strings, 800, <http://codeforces.com/problemset/problem/281/A>

Capitalization is writing a word with its first letter as a capital letter. Your task is to capitalize the given word.

Note, that during capitalization all the letters except the first one remains unchanged.

Input

A single line contains a non-empty word. This word consists of lowercase and uppercase English letters. The length of the word will not exceed 10^3 .

Output

Output the given word after capitalization.

```

line = input()
print(line[0].upper() + line[1:])

```

266A. Stones on the Table

implementation, 800, <http://codeforces.com/problemset/problem/266/A>

There are n stones on the table in a row, each of them can be red, green or blue. Count the minimum number of stones to take from the table so that any two neighboring stones had different colors. Stones in a row are considered neighboring if there are no other stones between them.

Input

The first line contains integer n ($1 \leq n \leq 50$) — the number of stones on the table.

The next line contains string s , which represents the colors of the stones. We'll consider the stones in the row numbered from 1 to n from left to right. Then the i -th character s equals "R", if the i -th stone is red, "G", if it's green and "B", if it's blue.

Output

Print a single integer — the answer to the problem.

```
n = int(input())
l = input()

nCount = 0
p = l[0]
for i in range(1,n):
    if l[i]==p:
        nCount += 1
    else:
        p = l[i]

print(nCount)
```

96A. Football

implementation/strings, 900, <http://codeforces.com/problemset/problem/96/A>

Petya loves football very much. One day, as he was watching a football match, he was writing the players' current positions on a piece of paper. To simplify the situation he depicted it as a string consisting of zeroes and ones. A zero corresponds to players of one team; a one corresponds to players of another team. If there are at least 7 players of some team standing one after another, then the situation is considered dangerous. For example, the situation 0010011011111101 is dangerous and 11110111011101 is not. You are given the current situation. Determine whether it is dangerous or not.

Input

The first input line contains a non-empty string consisting of characters "0" and "1", which represents players. The length of the string does not exceed 100 characters. There's at least one player from each team present on the field.

Output

Print "YES" if the situation is dangerous. Otherwise, print "NO".

```
l = input()

n_0 = 0
n_1 = 0
for c in l:
    if c=='0':
        n_0 += 1
        if n_0==7: break
        n_1 = 0
    else:
        n_1 += 1
```

```

        if n_1==7: break
        n_0 = 0

if n_0==7 or n_1==7:
    print('YES')
else:
    print('NO')

```

```

s=input()
print(['NO','YES'] ['0'*7 in s or '1'*7 in s])

```

615A. Bulbs

implementation, 800, <http://codeforces.com/contest/615/problem/A>

Vasya wants to turn on Christmas lights consisting of m bulbs. Initially, all bulbs are turned off. There are n buttons, each of them is connected to some set of bulbs. Vasya can press any of these buttons. When the button is pressed, it turns on all the bulbs it's connected to. Can Vasya light up all the bulbs?

If Vasya presses the button such that some bulbs connected to it are already turned on, they do not change their state, i.e. remain turned on.

Input

The first line of the input contains integers n and m ($1 \leq n, m \leq 100$) — the number of buttons and the number of bulbs respectively.

Each of the next n lines contains x_i ($0 \leq x_i \leq m$) — the number of bulbs that are turned on by the i -th button, and then x_i numbers y_{ij} ($1 \leq y_{ij} \leq m$) — the numbers of these bulbs.

Output

If it's possible to turn on all m bulbs print "YES", otherwise print "NO".

```

n, m = map(int, input().split())
s = set()
for _ in range(n):
    s.update(input().split()[1:])

print(['NO', 'YES'] [len(s)==m])

```

236A. Boy or Girl

brute force/implementation/strings, 800, <https://codeforces.com/problemset/problem/236/A>

Those days, many boys use beautiful girls' photos as avatars in forums. So it is pretty hard to tell the gender of a user at the first glance. Last year, our hero went to a forum and had a nice chat with a beauty (he thought so). After that they talked very often and eventually they became a couple in the network.

But yesterday, he came to see "her" in the real world and found out "she" is actually a very strong man! Our hero is very sad and he is too tired to love again now. So he came up with a way to recognize users' genders by their user names.

This is his method: if the number of distinct characters in one's user name is odd, then he is a male, otherwise she is a female. You are given the string that denotes the user name, please help our hero to determine the gender of this user by his method.

Input

The first line contains a non-empty string, that contains only lowercase English letters — the user name. This string contains at most 100 letters.

Output

If it is a female by our hero's method, print "CHAT WITH HER!" (without the quotes), otherwise, print "IGNORE HIM!" (without the quotes).

```
s = input()
fq = 26*[0]

for c in s:
    fq[ord(c) - ord('a')] += 1

nCount = 0
for i in range(26):
    if fq[i] != 0:
        nCount += 1

if nCount % 2 == 1:
    print('IGNORE HIM!')
else:
    print('CHAT WITH HER!')
```

```
s = set()
s.update(input())
if len(s) % 2 == 1:
    print('IGNORE HIM!')
else:
    print('CHAT WITH HER!')
```

update Method:

This method is used to return the union of a set and the set of elements from one or more iterable like string, list, set. It is very similar to **union()** method, with difference is that where union() method create and return a new set, containing all the elements (distinct) present in all the iterables, update() method updates the set on which this method is called with all the distinct elements present in all the iterables.

69A. Young Physicist

implementation/math, 1000, <https://codeforces.com/problemset/problem/69/A>

A guy named Vasya attends the final grade of a high school. One day Vasya decided to watch a match of his favorite hockey team. And, as the boy loves hockey very much, even more than physics, he forgot to do the homework. Specifically, he forgot to complete his physics tasks. Next day the teacher got very angry at Vasya and decided to teach him a lesson. He gave the lazy student a seemingly easy task: You are given an idle body in space and the forces that affect it. The body can be considered as a material point with coordinates (0; 0; 0). Vasya had only to answer whether it is in equilibrium. "Piece of cake" — thought Vasya, we need only to check if the sum of all vectors is equal to 0. So, Vasya began to solve the problem. But later it turned out that there can be lots and lots of these forces, and Vasya can not cope without your help. Help him. Write a program that determines whether a body is idle or is moving by the given vectors of forces.

Input

The first line contains a positive integer n ($1 \leq n \leq 100$), then follow n lines containing three integers each: the x_i coordinate, the y_i coordinate and the z_i coordinate of the force vector, applied to the body ($-100 \leq x_i, y_i, z_i \leq 100$).

Output

Print the word "YES" if the body is in equilibrium, or the word "NO" if it is not.

```
n = int(input())
x = 0
y = 0
z = 0
while n>0:
    n -= 1
    f = [int(i) for i in input().split()]
    x += f[0]
    y += f[1]
    z += f[2]

if x==y==z==0:
    print('YES')
else:
    print('NO')
```

25A. IQ test

brute force, 1300, <http://codeforces.com/problemset/problem/25/A>

Bob is preparing to pass IQ test. The most frequent task in this test is to find out which one of the given n numbers differs from the others. Bob observed that one number usually differs from the others in evenness. Help Bob — to check his answers, he needs a program that among the given n numbers finds one that is different in evenness.

Input

The first line contains integer n ($3 \leq n \leq 100$) — amount of numbers in the task. The second line contains n space-separated natural numbers, not exceeding 100. It is guaranteed, that exactly one of these numbers differs from the others in evenness.

Output

Output index of number that differs from the others in evenness. Numbers are numbered from 1 in the input order.

```

n = int(input())
l = [int(x) for x in input().split()]

cnt_even = 0

for i in range(n):
    if l[i]%2==0:
        cnt_even += 1

if cnt_even == 1:
    for i in range(n):
        if l[i]%2==0:
            print(i+1)
            break
else:
    for i in range(n):
        if l[i]%2 != 0:
            print(i+1)
            break

```

```

n = int(input())
l = input().split()
m = ''
for i in range(n):
    m = m + str(int(l[i])%2)

if m.count('1')==1:
    print(int(m.index('1'))+1)
else:
    print(int(m.index('0'))+1)

```

周晋飞

```

useless = input()
a = [int(x)%2 for x in input().split()]
print(a.index(sum(a)==1)+1)

```

122A. Lucky Division

brute force, number theory, 1000, <https://codeforces.com/problemset/problem/122/A>

Petya loves lucky numbers. Everybody knows that lucky numbers are positive integers whose decimal representation contains only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.

Petya calls a number almost lucky if it could be evenly divided by some lucky number. Help him find out if the given number n is almost lucky.

Input

The single line contains an integer n ($1 \leq n \leq 1000$) — the number that needs to be checked.

Output

In the only line print "YES" (without the quotes), if number n is almost lucky. Otherwise, print "NO" (without the quotes).

```
n = int(input())

for i in {4,7,47,74,447,474,477,747,774}:
    if n%i == 0:
        print('YES')
        break
else:
    print('NO')
```

2020fall-cs101-赵春源

```
def check(x):
    s = str(x)
    for c in s:
        if c != '4' and c != '7':
            return False
    return True

n = int(input())
for i in range(1, n+1):
    if n%i == 0 and check(i) == True:
        print('YES')
        break
else:
    print('NO')
```

```
def check(x):
    s = str(x)
    return [False, True][s.count('4') + s.count('7') == len(s)]

n = int(input())
for i in range(1, n+1):
    if n%i == 0 and check(i) == True:
        print('YES')
        break
else:
    print('NO')
```

用len函数是多么简单好用，学到了~

```

n = int(input())
luckynumbers = []
for i in range(1, 1002):
    s = str(i)
    if s.count('4') + s.count('7') == len(s):
        luckynumbers.append(i)

b = 0
for i in luckynumbers:
    if n%i == 0:
        b = 1
print(['NO', 'YES'][b])

```

any(x)判断x对象是否为空对象，如果都为空、0、false，则返回false，如果不都为空、0、false，则返回true

all(x)如果all(x)参数x对象的所有元素不为0、"、False或者x为空对象，则返回True，否则返回False

```

n = int(input())
print('NO' if all([n%i for i in (4,7,47,74,447,474,477,747,774)]) else 'YES')

```

723A. The New Year: Meeting Friends

implementation, math, sorting, 800, <https://codeforces.com/problemset/problem/723/A>

There are three friend living on the straight line Ox in Lineland. The first friend lives at the point x_1 , the second friend lives at the point x_2 , and the third friend lives at the point x_3 . They plan to celebrate the New Year together, so they need to meet at one point. What is the minimum total distance they have to travel in order to meet at some point and celebrate the New Year?

It's guaranteed that the optimal answer is always integer.

Input

The first line of the input contains three **distinct** integers x_1 , x_2 and x_3 ($1 \leq x_1, x_2, x_3 \leq 100$) — the coordinates of the houses of the first, the second and the third friends respectively.

Output

Print one integer — the minimum total distance the friends need to travel in order to meet together.

```

x = list(map(int, input().split()))
x.sort()
print(x[-1] - x[0])

```

705A. Hulk

implementation, 800, <https://codeforces.com/problemset/problem/705/A>

Dr. Bruce Banner hates his enemies (like others don't). As we all know, he can barely talk when he turns into the incredible Hulk. That's why he asked you to help him to express his feelings.

Hulk likes the Inception so much, and like that his feelings are complicated. They have n layers. The first layer is hate, second one is love, third one is hate and so on...

For example if $n = 1$, then his feeling is "I hate it" or if $n = 2$ it's "I hate that I love it", and if $n = 3$ it's "I hate that I love that I hate it" and so on.

Please help Dr. Banner.

Input

The only line of the input contains a single integer n ($1 \leq n \leq 100$) — the number of layers of love and hate.

Output

Print Dr.Banner's feeling in one line.

2020fall-cs101-郭冠廷,

```
say = []
for i in range(int(input())):
    say.append(['I hate', 'I love'][i % 2])
print(" that ".join(say), end=" it\n")
```

```
n = int(input())

l = ""
for i in range(1,n):
    if i%2 == 0:
        l += " that I hate"
    else:
        l += " that I love"

print("I hate" + l + " it")
```

2020fall-cs101-成泽凯, 解题思路:

"I hate that"和"I love that"用 while来输出, 最后根据 n 的奇偶来判断输出"I love it"还是"I hate it"

```
n = int(input())
a = n
while n > 1:
    print("I hate that", end=" ")
    n -= 1
    if n > 1:
        print("I love that", end=" ")
        n -= 1

if a%2 == 0:
    print("I love it")
else:
    print("I hate it")
```

200B. Drinks

implementation, math, 800, <https://codeforces.com/problemset/problem/200/B>

Little Vasya loves orange juice very much. That's why any food and drink in his kitchen necessarily contains orange juice. There are n drinks in his fridge, the volume fraction of orange juice in the i -th drink equals p_i percent.

One day Vasya decided to make himself an orange cocktail. He took equal proportions of each of the n drinks and mixed them. Then he wondered, how much orange juice the cocktail has.

Find the volume fraction of orange juice in the final drink.

Input

The first input line contains a single integer n ($1 \leq n \leq 100$) — the number of orange-containing drinks in Vasya's fridge. The second line contains n integers p_i ($0 \leq p_i \leq 100$) — the volume fraction of orange juice in the i -th drink, in percent. The numbers are separated by a space.

Output

Print the volume fraction in percent of orange juice in Vasya's cocktail. The answer will be considered correct if the absolute or relative error does not exceed 10^{-4} .

```
n=int(input())
p = list(map(int,input().split()))
print(sum(p)/n)
```

492B. Vanya and Lanterns

binary search/implementation/math/sortings, 1200, <https://codeforces.com/problemset/problem/492/B>

Vanya walks late at night along a straight street of length l , lit by n lanterns. Consider the coordinate system with the beginning of the street corresponding to the point 0, and its end corresponding to the point l . Then the i -th lantern is at the point a_i . The lantern lights all points of the street that are at the distance of at most d from it, where d is some positive number, common for all lanterns.

Vanya wonders: what is the minimum light radius d should the lanterns have to light the whole street?

Input

The first line contains two integers n, l ($1 \leq n \leq 1000, 1 \leq l \leq 10^9$) — the number of lanterns and the length of the street respectively.

The next line contains n integers a_i ($0 \leq a_i \leq l$). Multiple lanterns can be located at the same point. The lanterns may be located at the ends of the street.

Output

Print the minimum light radius d , needed to light the whole street. The answer will be considered correct if its absolute or relative error doesn't exceed 10^{-9} .

```

n,l=map(int,input().split())
a=list(map(int,input().split()))
ans=0
a.sort()
for i in range(n-1):
    ans = max(ans, (a[i+1]-a[i])/2)
print(max(ans, a[0], l-a[-1]))

```

508A. Pasha and Pixels

brute force, 1100, <http://codeforces.com/problemset/problem/508/A>

Pasha loves his phone and also putting his hair up... But the hair is now irrelevant.

Pasha has installed a new game to his phone. The goal of the game is following. There is a rectangular field consisting of n row with m pixels in each row. Initially, all the pixels are colored white. In one move, Pasha can choose any pixel and color it black. In particular, he can choose the pixel that is already black, then after the boy's move the pixel does not change, that is, it remains black. Pasha loses the game when a 2×2 square consisting of black pixels is formed.

Pasha has made a plan of k moves, according to which he will paint pixels. Each turn in his plan is represented as a pair of numbers i and j , denoting respectively the row and the column of the pixel to be colored on the current move.

Determine whether Pasha loses if he acts in accordance with his plan, and if he does, on what move the 2×2 square consisting of black pixels is formed.

Input

The first line of the input contains three integers n, m, k ($1 \leq n, m \leq 1000$, $1 \leq k \leq 10^5$) — the number of rows, the number of columns and the number of moves that Pasha is going to perform.

The next k lines contain Pasha's moves in the order he makes them. Each line contains two integers i and j ($1 \leq i \leq n$, $1 \leq j \leq m$), representing the row number and column number of the pixel that was painted during a move.

Output

If Pasha loses, print the number of the move when the 2×2 square consisting of black pixels is formed.

If Pasha doesn't lose, that is, no 2×2 square consisting of black pixels is formed during the given k moves, print 0.

练习加保护圈

```

# http://codeforces.com/contest/508/submission/44603553
n,m,k = map(int, input().split())
mx = [(m+2)*[0] for i in range(n+2)]

# if square 2x2 formed from black cells appears, and
# cell (i,j) will upper-left, upper-right, bottom-left
# or bottom-right of this squares.

def square_check(i,j):
    if mx[i][j+1] and mx[i+1][j] and mx[i+1][j+1]:

```

```

        return True
    if mx[i][j-1] and mx[i+1][j-1] and mx[i+1][j]:
        return True
    if mx[i-1][j] and mx[i-1][j+1] and mx[i][j+1]:
        return True
    if mx[i-1][j-1] and mx[i-1][j] and mx[i][j-1]:
        return True
    return False

for i in range(k):
    x,y = map(int, input().split())
    mx[x][y] = 1
    if square_check(x,y):
        print(i+1)
        break
else:
    print(0)

```

456A. Laptops

sortings, 1100, <https://codeforces.com/problemset/problem/456/A>

One day Dima and Alex had an argument about the price and quality of laptops. Dima thinks that the more expensive a laptop is, the better it is. Alex disagrees. Alex thinks that there are two laptops, such that the price of the first laptop is less (strictly smaller) than the price of the second laptop but the quality of the first laptop is higher (strictly greater) than the quality of the second laptop.

Please, check the guess of Alex. You are given descriptions of n laptops. Determine whether two described above laptops exist.

Input

The first line contains an integer n ($1 \leq n \leq 10^5$) — the number of laptops.

Next n lines contain two integers each, a_i and b_i ($1 \leq a_i, b_i \leq n$), where a_i is the price of the i -th laptop, and b_i is the number that represents the quality of the i -th laptop (the larger the number is, the higher is the quality).

All a_i are distinct. All b_i are distinct.

Output

If Alex is correct, print "Happy Alex", otherwise print "Poor Alex" (without the quotes).

```

n = int(input())
s = [[int(x) for x in input().split()] for _ in range(n)]
s.sort(reverse=True)

for i in s[1:]:
    if i[1] > s[0][1]:
        print('Happy Alex')
        break

    s[0][1] = i[1]
else:
    print('Poor Alex')

```


OPTIONAL PROBLEMS

368B. Sereja and Suffixes

data structures/dp, 1100, <https://codeforces.com/problemset/problem/368/B>

Sereja has an array a , consisting of n integers a_1, a_2, \dots, a_n . The boy cannot sit and do nothing, he decided to study an array. Sereja took a piece of paper and wrote out m integers l_1, l_2, \dots, l_m ($1 \leq l_i \leq n$). For each number l_i he wants to know how many distinct numbers are staying on the positions $l_i, l_i + 1, \dots, n$. Formally, he want to find the number of distinct numbers among $a_{l_i}, a_{l_i + 1}, \dots, a_n$.

Sereja wrote out the necessary array elements but the array was so large and the boy was so pressed for time. Help him, find the answer for the described question for each l_i .

Input

The first line contains two integers n and m ($1 \leq n, m \leq 10^5$). The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^5$) — the array elements.

Next m lines contain integers l_1, l_2, \dots, l_m . The i -th line contains integer l_i ($1 \leq l_i \leq n$).

Output

Print m lines — on the i -th line print the answer to the number l_i .

Examples

input

```
10 10
1 2 3 4 1 2 3 4 100000 99999
1
2
3
4
5
6
7
8
9
10
```

output

6
6
6
6
6
5
4
3
2
1

要求数据结构实现一次，dp实现一次

DP方法：

2020fall-cs101，郭冠廷。本题的思路不困难，只需反序 dp 即可（甚至看不出 dp）。为了节省时间，应用到了两个窍门：1) 将已经读取的数据投入到 set 中，这样能避免列表查找的缓速。2) 将答案储存在一个列表中，同时输出。二者都是以空间换时间。

```
n,m = map(int, input().split())
n1 = [int(x) for x in input().split()]

out = 0
dp = []

tset = set()
for i in reversed(n1):
    if i not in tset:
        out += 1
        tset.add(i)
    dp.append(out)

for _ in range(m):
    pos = len(n1) - int(input())
    print(dp[pos])
```

2020fall-cs101，王康安。最开始用 in 去判断，tle 了。于是开一个到一万的桶，空间换时间，成功 AC。

```
n,m = map(int, input().split())
seq = [int(x) for x in input().split()]
dp = [0]*(n-1)+[1]
barrel = [False]*(100000+1)
barrel[seq[n-1]] = True
for i in reversed(range(n-1)):
    if barrel[seq[i]]:
        dp[i] = dp[i+1]
    else:
        dp[i] = dp[i+1]+1
        barrel[seq[i]] = True
for i in range(m):
    print(dp[int(input())-1])
```

data structures方法:

2020fall-cs101, 蓝克轩。解题思路: 先用dictionary记录每个元素数量, 然后从第一个元素开始, 每一个元素从dictionary减一, 如果在dictionary对应对应的因素是0就去掉, dictionary的长度即是不同元素的数量。

```
n, m = map(int, input().split())
a = list(map(int, input().split()))
adic = {}
for i in range(n):
    if a[i] in adic:
        adic[a[i]] += 1
    else:
        adic[a[i]] = 1

ans = [0]*n
for i in range(n):
    if a[i] in adic:
        if adic[a[i]] == 1:
            adic.pop(a[i])
            ans[i] += 1
        else:
            adic[a[i]] -= 1
    ans[i] += len(adic)

for i in range(m):
    print(ans[int(input()) - 1])
```

313B. Ilya and Queries

dp, 1300, <https://codeforces.com/contest/313/problem/B>

Ilya the Lion wants to help all his friends with passing exams. They need to solve the following problem to pass the IT exam.

You've got string $s = s_1s_2...s_n$ (n is the length of the string), consisting only of characters "." and "#" and m queries. Each query is described by a pair of integers l_i, r_i ($1 \leq l_i < r_i \leq n$). The answer to the query l_i, r_i is the number of such integers i ($l_i \leq i < r_i$), that $s_i = s_{i+1}$.

Ilya the Lion wants to help his friends but is there anyone to help him? Help Ilya, solve the problem.

Input

The first line contains string s of length n ($2 \leq n \leq 10^5$). It is guaranteed that the given string only consists of characters "." and "#".

The next line contains integer m ($1 \leq m \leq 10^5$) — the number of queries. Each of the next m lines contains the description of the corresponding query. The i -th line contains integers l_i, r_i ($1 \leq l_i < r_i \leq n$).

Output

Print m integers — the answers to the queries in the order in which they are given in the input.

Examples

sample1 input

```
.....  
4  
3 4  
2 3  
1 6  
2 6
```

sample1 output

```
1  
1  
5  
4
```

sample2 input

```
#. .###  
5  
1 3  
5 6  
1 5  
3 6  
3 4
```

sample2 output

```
1  
1  
2  
2  
0
```

2020fall-cs101, 郭佩妤

```
s = input()  
out = 0  
dp = [0]  
for i in range(1, len(s)):  
    if s[i] == s[i-1]:  
        out += 1  
        dp.append(out)  
  
m = int(input())  
for i in range(m):  
    left, right = map(int, input().split())  
    print(dp[right-1] - dp[left-1])
```

2020fall-cs101, 李元锋

先把所有长度从 1 到 i 记录一遍放在 list 里，之后对输入值做减法即可。这里也是需要把答案放在一个 list 里全部输出，不然会超时。

```

s = [str(i) for i in input()]
ans = [0]
num = 0
answer = []
for i in range(1, len(s)):
    if s[i]==s[i-1]:
        num += 1
    ans.append(num)

for i in range(int(input())):
    l,r = map(int, input().split())
    answer.append(str(ans[r-1] - ans[l-1]))

print('\n'.join(answer))

```

706B. Interesting drink

binary search/dp/implementation, 1100, <https://codeforces.com/problemset/problem/706/B>

Vasiliy likes to rest after a hard work, so you may often meet him in some bar nearby. As all programmers do, he loves the famous drink "Beecola", which can be bought in n different shops in the city. It's known that the price of one bottle in the shop i is equal to x_i coins.

Vasiliy plans to buy his favorite drink for q consecutive days. He knows, that on the i -th day he will be able to spent m_i coins. Now, for each of the days he want to know in how many different shops he can buy a bottle of "Beecola".

Input

The first line of the input contains a single integer n ($1 \leq n \leq 100\,000$) — the number of shops in the city that sell Vasiliy's favourite drink.

The second line contains n integers x_i ($1 \leq x_i \leq 100\,000$) — prices of the bottles of the drink in the i -th shop.

The third line contains a single integer q ($1 \leq q \leq 100\,000$) — the number of days Vasiliy plans to buy the drink.

Then follow q lines each containing one integer m_i ($1 \leq m_i \leq 10^9$) — the number of coins Vasiliy can spent on the i -th day.

Output

Print q integers. The i -th of them should be equal to the number of shops where Vasiliy will be able to buy a bottle of the drink on the i -th day.

Example

input

```
5
3 10 8 6 11
4
1
10
3
11
```

output

```
0
4
1
5
```

Note

On the first day, Vasiliy won't be able to buy a drink in any of the shops.

On the second day, Vasiliy can buy a drink in the shops 1, 2, 3 and 4.

On the third day, Vasiliy can buy a drink only in the shop number 1.

Finally, on the last day Vasiliy can buy a drink in any shop.

要求二分实现一次，dp实现一次

binary search实现

```
import bisect

n = int(input())
x = [int(_) for _ in input().split()]
x.sort()

for _ in range(int(input())):
    m = int(input())
    print(bisect.bisect_right(x, m))
```

DP实现

```
input()
x = [int(_) for _ in input().split()]
nm = max(x)
c = [0]*(nm+1)
for i in x:
    c[i] += 1

for i in range(2, nm+1):
    c[i] += c[i-1]

#ans = []
for _ in range(int(input())):
    m = int(input())
```

```

if m > nm:
    print(c[nm])
else:
    print(c[m])
#ans.append(c[m])

#print('\n'.join(map(str,ans)))

```

2020fall-cs101, 成泽恺

706B 思路：暴力超时，用二分查找或者动态规划二分查找：一般二分查找会返回数的位置或者找不到，但这个题只需要找到价格中小于等于输入和大于等于输入的分界线即可，所以查找上界 left 和 right，在 while 循环中取 left>=mid 时right=mid-1，最后结果就是上界 right+1。

```

n = int(input())
shop = sorted(list(map(int,input().split())))
m = int(input())
for i in range(m):
    a = int(input())
    l = 0
    r = n-1
    while l<=r:
        mid = (l+r)//2
        if a < shop[mid]:
            r = mid-1
        elif a >= shop[mid]:
            l = mid + 1
    print(r+1)

```

2020fall-cs101, 黄旭

解题思路：这道题试了一下用 bisect库做，觉得挺方便的

```

import bisect
input()
lt=sorted([int(i) for i in input().split()])
for i in range(int(input())):
    print(bisect.bisect(lt,int(input())))

```

2020fall-cs101, 邹京驰

思路：让区间[a,b]不断折半，且始终满足 $x[a] \leq p < x[b]$ ；直到 $a = b - 1$ 。a就是我“刚好吃的起”的那个餐馆的位次，也即我吃得起的餐馆的总数量。binary search 也是第一次见。思考怎么实现时的第一反应：“这不就是高数里的闭区间套吗？”于是按自己的思路很快写出来了，并且一次submitAC. 很满意。

```

def search(t):
    a = 0
    b = n
    while a != b-1:
        temp = (a+b)//2
        if x[temp] > t:

```

```

        b = temp
    else:
        a = temp

    return a

n = int(input())
x = [int(x) for x in input().split()]
q = int(input())
x.sort()
x.insert(0, 0)

for i in range(q):
    p = int(input())
    if p >= x[n]:
        print(n)
    else:
        print(search(p))

```

1425A. Arena of Greed

games/greedy, 1400, <https://codeforces.com/problemset/problem/1425/A>

Lately, Mr. Chanek frequently plays the game **Arena of Greed**. As the name implies, the game's goal is to find the greediest of them all, who will then be crowned king of Compfestnesia.

The game is played by two people taking turns, where Mr. Chanek takes the first turn. Initially, there is a treasure chest containing N gold coins. The game ends if there are no more gold coins in the chest. In each turn, the players can make one of the following moves:

- Take one gold coin from the chest.
- Take half of the gold coins on the chest. This move is only available if the number of coins in the chest is even.

Both players will try to maximize the number of coins they have. Mr. Chanek asks your help to find the maximum number of coins he can get at the end of the game if both he and the opponent plays optimally.

Input

The first line contains a single integer T ($1 \leq T \leq 10^5$) denotes the number of test cases.

The next T lines each contain a single integer N ($1 \leq N \leq 10^{18}$).

Output

T lines, each line is the answer requested by Mr. Chanek.

思路：为了获取最多的石子数量：

1. 数量为奇数时：只能取1个，然后对手进入情况2，我们只能取剩下的；
2. 数量为偶数时：为了尽可能最大化所能取的石子数量，我们尽可能使得对手只能取1个，即使得对手取时数量为奇数；同时使得我们取石子时数量为偶数。
为了实现这个情况，判断一下当前石子数量的一半是否为奇数，如果是，我们就取一半；如果不

是，我们就取一个，对应的，对手也只能取一个，之后所得到的偶数的一般必然是个奇数。证明略。

3. 此外1和4是特殊情况，需要特判一下。

PyPy 3 AC. Python 3 Time limit exceeded on Test2.

```
#input=__import__('sys').stdin.readline
ans = []
def solve(n):

    f = s = 0 # To distinguish between first and second hands.
    fs = True

    if n & 1:
        n -= 1
        fs = False

    while n:
        if n == 4:
            f += 3
            s += 1
            n = 0 # Special Judge
        elif (n//2) & 1: # The First Situation
            f += n//2
            s += 1
            n = (n//2) - 1;
        else: #The Second Situation
            f += 1
            s += 1
            n -= 2
        #print( [s+1, f][fs] )
        ans.append( [s+1, f][fs] )

    coins = []
    for _ in range(int(input())):
        coins.append(int(input()))

    for i in coins:
        if i==1:
            ans.append(1)
            #print(1)
        else:
            solve(i)

    print('\n'.join(map(str, ans)))
```

1443C. The Delivery Dilemma

binary search/greedy/sortings, 1400, <https://codeforces.com/problemset/problem/1443/C>

Petya is preparing for his birthday. He decided that there would be n different dishes on the dinner table, numbered from 1 to n . Since Petya doesn't like to cook, he wants to order these dishes in restaurants.

Unfortunately, all dishes are prepared in different restaurants and therefore Petya needs to pick up his orders from n different places. To speed up this process, he wants to order courier delivery at some restaurants. Thus, for each dish, there are two options for Petya how he can get it:

- the dish will be delivered by a courier from the restaurant i , in this case the courier will arrive in a_i minutes,
- Petya goes to the restaurant i on his own and picks up the dish, he will spend b_i minutes on this.

Each restaurant has its own couriers and they start delivering the order at the moment Petya leaves the house. In other words, all couriers work in parallel. Petya must visit all restaurants in which he has not chosen delivery, he does this consistently.

For example, if Petya wants to order $n=4$ dishes and $a=[3,7,4,5]$, and $b=[2,1,2,4]$, then he can order delivery from the first and the fourth restaurant, and go to the second and third on your own. Then the courier of the first restaurant will bring the order in 3 minutes, the courier of the fourth restaurant will bring the order in 5 minutes, and Petya will pick up the remaining dishes in $1+2=3$ minutes. Thus, in 5 minutes all the dishes will be at Petya's house.

Find the minimum time after which all the dishes can be at Petya's home.

Input

The first line contains one positive integer t ($1 \leq t \leq 2 \cdot 10^5$) — the number of test cases. Then t test cases follow.

Each test case begins with a line containing one integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of dishes that Petya wants to order.

The second line of each test case contains n integers $a_1 \dots a_n$ ($1 \leq a_i \leq 10^9$) — the time of courier delivery of the dish with the number i .

The third line of each test case contains n integers $b_1 \dots b_n$ ($1 \leq b_i \leq 10^9$) — the time during which Petya will pick up the dish with the number i .

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case output one integer — the minimum time after which all dishes can be at Petya's home.

2020fall-cs101-张聪, 思路: 由于 delivery 是并行的, pick 是串行的, 很自然地想到 delivery 应该优先。所以构建二维数组并对其根据 delivery 时间排序, 然后用 greedy 算法思想, 找到最小的并行时间, 要求其能够覆盖 delivery 时间更长的 dish 的串行时间之和。

2020fall-cs101-施惟明, 思路: 1) 听闻此题 tle 主要来源于读入数据的耗时, 便考虑采用空间换时间的策略, 先一波读入所有数据, 再进行处理。2) 处理部分仍然是 greedy, 尽量自己取耗时长的外卖, 当自己取的时长和外卖送到的最慢时长相等时为最佳。复杂度 $O(n \log n)$ (上一次好像忘记考虑 sort 的复杂度了...)。

2020fall-cs101-成泽凯，思路：首先将每个菜的配送用时 a_i 和自取用时 b_i 配对，然后按配送用时 a_i 降序排。因为配送时每个餐馆同时送，所以实际上配送用时短的会被配送用时长的覆盖掉。从前向后遍历，并求自取用时的前缀和 $sum[j]$ ，当自取用时的前缀和比配送用时 a_j 大。

(由于Test9有 200000个输入，所以一个一个输出处理会超时，经施惟明，成泽恺等同学提醒后，修改了Python代码，使得只输出一次，成功 ac 了。)

2020fall-cs101-王君宇，当我写完高数作业看群，群里的大佬说 python常规方法可以 AC!! 我们平时做这种多组 test问题都是直接循环每次 print，但是也可以将每次结果存到一个列表中，最后一起 print，这样能大大加快速度。确实我对于 python中各个函数的运行速度还不够了解，这道题锻炼了我的程序思维，更让我加深了对 print函数的理解。当我看到自己的代码 AC的那一刻，这个下午的失败值得了!!!

```
n = int(input())
ans = []
for i in range(n):
    m = int(input())
    a = list(map(int, input().split()))
    b = list(map(int, input().split()))
    c = sorted(list(zip(a,b)), reverse=True)

    d=0
    for i in range(m):
        d += c[i][1]
        if d >= c[i][0]:
            d = max(c[i][0], d-c[i][1])
            break

    ans.append(d)

print('\n'.join(map(str, ans)))
```

```
T = int(input())
to_print = []
while T:
    T -= 1

    n = int(input())
    a = [int(x) for x in input().split()]
    b = [int(x) for x in input().split()]

    if n==1:
        to_print.append( min(a[0], b[0]) )
        #print(min(a[0], b[0]))
        continue

    ab = list(zip(a,b))
    ab.sort()

    k = n - 1
    suffix_sum = 0
    while suffix_sum <= ab[k][0] and k>=0:
        suffix_sum += ab[k][1]
        k -= 1
```

```

        #print( min(ab[k+1][0], suffix_sum))
        to_print.append( ( min(ab[k+1][0], suffix_sum)) )

    else:
        print('\n'.join(map(str, to_print)))

```

```

T = int(input())
to_print = []
while T:
    T -= 1
    n = int(input())
    a = [int(x) for x in input().split()]
    b = [int(x) for x in input().split()]

    if n==1:
        #print(min(a[0], b[0]))
        to_print.append( min(a[0], b[0]) )
        continue

    ab = list(zip(a,b))
    ab.sort( key = lambda x: (-x[0], x[1]))

    for i in range(1, n):
        ab[i] = ab[i][0], (ab[i][1]+ab[i-1][1])

    #print(ab)

    #print( max ( map(min, ab) ) )
    to_print.append( max ( map(min, ab) ) )

else:
    print('\n'.join(map(str, to_print)))

```

2020fall-cs101, 许浩哲, 1443c 可以用Python AC的另外一种方法。就是把nput换成sys.stdin.readline, 把print换成sys.stdout.write。 <https://codeforces.com/problemset/problem/1443/C>

```

import sys

n = int(input())
#n = int(sys.stdin.readline())
#ans = []
for i in range(n):
    #m = int(input())
    m = int(sys.stdin.readline())
    a = list(map(int,sys.stdin.readline().split()))
    b = list(map(int,sys.stdin.readline().split()))
    c = sorted(list(zip(a,b)),reverse=True)

    d=0
    for i in range(m):

```

```

        d += c[i][1]
        if d >= c[i][0]:
            d = max(c[i][0], d-c[i][1])
            break

    sys.stdout.write('{}\n'.format(d))
    #ans.append(d)

#print('\n'.join(map(str, ans)))

```

GNU C++17 Accepted

```

#include <bits/stdc++.h>
using namespace std;
pair <int,int> a[200001];
int main(){
    int t;
    cin >> t;
    while(t--){
        int n;
        cin>>n;
        int k=0,sum=0;
        for(k=1;k<=n;k++)cin>>a[k].first;
        for(k=1;k<=n;k++)cin>>a[k].second;
        sort(a+1,a+n+1);
        k=n;
        while(sum<=a[k].first &&k >=1){
            sum += a[k].second;
            k--;
        }
        cout << min(a[k+1].first, sum) << endl;
    }
    return 0;
}

```

189A. Cut Ribbon

brute force/dp, 1300, <https://codeforces.com/problemset/problem/189/A>

Polycarpus has a ribbon, its length is n . He wants to cut the ribbon in a way that fulfils the following two conditions:

- After the cutting each ribbon piece should have length a , b or c .
- After the cutting the number of ribbon pieces should be maximum.

Help Polycarpus and find the number of ribbon pieces after the required cutting.

Input

The first line contains four space-separated integers n , a , b and c ($1 \leq n, a, b, c \leq 4000$) — the length of the original ribbon and the acceptable lengths of the ribbon pieces after the cutting, correspondingly. The numbers a , b and c can coincide.

Output

Print a single number — the maximum possible number of ribbon pieces. It is guaranteed that at least one correct ribbon cutting exists.

Examples

input

```
5 5 3 2
```

output

```
2
```

input

```
7 5 5 2
```

output

```
2
```

Note

In the first example Polycarpus can cut the ribbon in such way: the first piece has length 2, the second piece has length 3.

In the second example Polycarpus can cut the ribbon in such way: the first piece has length 5, the second piece has length 2.

2020fall-cs101, 王君宇。这道题状态转移方程就是 $d[i] = \max(d[i-a], d[i-b], d[i-c]) + 1$, 其中初始量是 0. 到达一个新节点的方法有三种: +a、+b、+c, 选取最大增量即可, 思路十分精巧。

2020fall-cs101, 黄旭。找到递推公式 $dp[i] = \max(dp[i-a], dp[i-b], dp[i-c]) + 1$ 就好了。

```
n,a,b,c = map(int,input().split())
dp = [0]+[-4000]*4000
for i in range(1,n+1):
    dp[i] = max(dp[i-a], dp[i-b], dp[i-c]) + 1
print(dp[n])
```

2020fall-cs101, 赵春源。这是一个简单的DP思想, 我们让 f_i 等于把 i 按题意切开所得的最大段数, 我们让 $f_0 = 0$, 其他的位置为负无穷, 表示这个长度没法按题目的要求切开, 我们考虑如何用较小的 f 推出较大的 f , 当然是把a,b,c三种切法都试一遍, 最大长度就是 $\max(f[i-a], f[i-b], f[i-c]) + 1$

```

n, a, b, c = map(int, input().split())
f = (n+1) * [-10000]
f[0] = 0
for i in range(1, n+1):
    if i >= a:
        f[i] = max(f[i], f[i - a] + 1)
    if i >= b:
        f[i] = max(f[i], f[i - b] + 1)
    if i >= c:
        f[i] = max(f[i], f[i - c] + 1)
print(f[n])

```

2020fall-cs101, 成泽恺。一开始直接想暴力循环，三个片段a, b, c，一开始假设他可以分成i个a和j个b, i, j初始值都为0，while a_i和a_i+b_j比n小的时候循环判断能不能加c片段进去，如果可以用max函数拿到此刻的片段数目的最大值，直到最后输出num即可，但这样用python3会超时，pypy3可以过。

```

n,a,b,c = map(int, input().split())
num = 0
i = 0
j = 0
while a*i <= n:
    j = 0
    while b*j+a*i <= n:
        d = n - a*i - b*j
        if d%c==0 and a*i+b*j<=n:
            num = max(num, i+j+d//c)
        j += 1
    i += 1
print(num)

```

题目需要用到动态规划，这个问题相当于一个完全背包，背包容量是n，有重量为a, b, c的物品，物品价值都是1，求取在恰好装满背包的情况下价值最大。因为题目保证有解，开一个长度为n+1的列表，初始值为-9999（避免背包不被恰好装满的情况出现），容量为0的时候价值为0，容量为i的时候判断能不能放下重量为a, b, c的片段，如果能查看背包剩余容量i-a可以放多少，如果i-a不能被恰好填满，由于初始值是很大的负值，在装进去之后仍然是负值，不会影响最终结果。循环n次，最后得到最优解。

2020fall-cs101, 刘安澜。思路：看上去这个题和老师上课讲的dp是一个镜像关系的题目，所以就按照找硬币的dp思路写了最初始的一版。但是不同于找硬币必有一个解，剪丝带对于某些丝带长度是无法分割的，所以这也就造成了在某些不能分割的长度上输出错误的结果。所以对于这种情况需要我们将除0外所有的初始值都赋为一个很大的负数，这样就能很好避免不能分割的长度对于其它可分割长度答案的扰动。

```
def dpcut( lengthlist, l, maxcuts ):
    for i in range( l + 1 ):
        for j in [ c for c in lengthlist if c <= i ]:
            if maxcuts[i-j] + 1 > maxcuts[i]:
                maxcuts[i] = maxcuts[i-j] + 1
    return maxcuts[l]

l,a,b,c = map(int,input().split())
lengthlist = [a, b, c]
maxcuts = [0] + [-1000000]*l

print( dpcut( lengthlist, l, maxcuts ) )
```

580C. Kefa and Park

dfs and similar/graphs/trees, 1500, <http://codeforces.com/problemset/problem/580/C>

Kefa decided to celebrate his first big salary by going to the restaurant.

He lives by an unusual park. The park is a rooted tree consisting of n vertices with the root at vertex 1. Vertex 1 also contains Kefa's house. Unfortunately for our hero, the park also contains cats. Kefa has already found out what are the vertices with cats in them.

The leaf vertices of the park contain restaurants. Kefa wants to choose a restaurant where he will go, but unfortunately he is very afraid of cats, so there is no way he will go to the restaurant if the path from the restaurant to his house contains more than m **consecutive** vertices with cats.

Your task is to help Kefa count the number of restaurants where he can go.

Input

The first line contains two integers, n and m ($2 \leq n \leq 10^5$, $1 \leq m \leq n$) — the number of vertices of the tree and the maximum number of consecutive vertices with cats that is still ok for Kefa.

The second line contains n integers a_1, a_2, \dots, a_n , where each a_i either equals to 0 (then vertex i has no cat), or equals to 1 (then vertex i has a cat).

Next $n - 1$ lines contains the edges of the tree in the format " $x_i y_i$ " (without the quotes) ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$), where x_i and y_i are the vertices of the tree, connected by an edge.

It is guaranteed that the given set of edges specifies a tree.

Output

A single integer — the number of distinct leaves of a tree the path to which from Kefa's home contains at most m consecutive vertices with cats.

常规做法应该是图的遍历。可以参考bfs: <https://www.codespeedy.com/breadth-first-search-algorithm-in-python/>

```
n,m = [int(i) for i in input().split()]
cat = [0]+[int(i) for i in input().split()]
d = {}
t = 1
for i in range(n-1):
```



```

x,y = [int(_) for _ in input().split()]
# d.setdefault(x, []).append(y)
try:
    d[x].append(y)
except:
    d[x] = [y]
# d.setdefault(y, []).append(x)
try:
    d[y].append(x)
except:
    d[y] = [x]

rec = [(1,0,1)]
cnt = 0
while len(rec) != 0:
    i,c,prev = rec.pop()
    if cat[i]:
        c += 1
    else:
        c = 0
    if c > m:
        continue
    if i != 1 and len(d[i]) == 1:
        cnt += 1
        continue
    for j in d[i]:
        if j == prev:
            continue
        rec.append((j,c,i))
print(cnt)

```

466C. Number of Ways

binary search/brute force/data structures/dp/two pointers, 1700, <https://codeforces.com/problemset/problem/466/C>

You've got array $a[1], a[2], \dots, a[n]$, consisting of n integers. Count the number of ways to split all the elements of the array into three contiguous parts so that the sum of elements in each part is the same.

More formally, you need to find the number of such pairs of indices i, j ($2 \leq i \leq j \leq n - 1$), that

$$\sum_{k=1}^{i-1} a_k = \sum_{k=i}^j a_k = \sum_{k=j+1}^n a_k$$

Input

The first line contains integer n ($1 \leq n \leq 5 \cdot 10^5$), showing how many numbers are in the array. The second line contains n integers $a[1], a[2], \dots, a[n]$ ($|a[i]| \leq 10^9$) — the elements of array a .

Output

Print a single integer — the number of ways to split the array into three parts with the same sum.

2020fall-cs1010-汤建浩，思路：难点在于总和能除以三的部分。由于是分成连续的三部分，每部分一样，只需要对列表一项项地加下去。其中一些节点非常重要，如前缀和达到1/3 和达到2/3，此两项都要满足，且存在前缀和达到1/3先于前缀和达到2/3的情况，才能分成连续三份。出现一次前缀和达到1/3 不能说明什么，但我们要记录下来s++，在它之后出现前缀和2/3，我们就把这个情况ans += s

2020fall-cs101-王君宇，思路：这道题的最大坑点就是分割是有序的...我因为没认真审题而半天毫无头绪。既然是有序的，我们可以遍历逐个求和。如果总和不是 3 的倍数显然不能分割，否则我们先得到 sum/3，之后是 2*sum/3。因此我们可以分成两步。当我们得到第一个分割之后，每得到一种便相应将二级种数加一。得到第二个分割后就将之前分割的总种数加起来。这里运用了类似于乘法原理的分布计数思想。另外，我们求和的终点应当是倒数第二个元素，防止 sum==0时总种数出错。

2020fall-cs101-黄旭，思路：将列表中的元素进行累加，由于要将其三等分，所以计算总和的三分之一以及三分之二，并应该将能得到三分之一的方法数乘以能得到三分之二的方法数从而得到方法总数，方法能想到就不难，不然可能要想很久很久（比如昨晚做到很晚）

双指针，同向右走，前缀和2/3的指针一定在前缀和1/3的右侧，所以正确。

例如：

5

3 3 -3 3 3

双指针向右走的过程

ans=1 ans+=s
=> =>
3 3 -3 3 3
-> -> ->
s=1 s=2

```
n = int(input())
a = [int(i) for i in input().split()]
s = sum(a)
x = 0
if s%3 == 0:
    d1 = s/3
    d2 = 2*d1
    z = t = 0
    for i in range(n-1):
        z += a[i]

        if z == d2:
            x += t
        if z == d1:
            t += 1
print(x)
```

解题思路：前缀和优化。当前缀和达到1/3时s++，达到2/3时候 ans += s

```
n = int(input())
a = [0]
a.extend([int(x) for x in input().split()])

prefix_sum = [0]*(n+1)

for i in range(1, n+1):
    prefix_sum[i] = prefix_sum[i-1] + a[i]

if (prefix_sum[n]%3) != 0:
    print(0)
else:
    s = ans = 0
    for i in range(1, n+1):
        if i>1 and i<n and prefix_sum[i]==(prefix_sum[n]*2//3):
            ans += s

        if prefix_sum[i] == (prefix_sum[n]//3):
            s += 1

    print(ans)
```

解题思路：tutorial, <https://codeforces.com/blog/entry/13758>

First of all, notice that if sum of all elements is equal S then sum of each of three parts is equal $S/3$. Therefore, if S is not divided by 3 — then answer is 0.

Otherwise, let's iterate the end of first part i ($1 \leq i \leq n-2$) and if sum of $1..i$ elements is equal $S/3$ then it means that we have to add to the answer the amount of such j ($i+1 < j$) that the sum of elements from j -th to n -th also equals $S/3$.

Let's create an array $cnt[]$, where $cnt[i]$ equals 1, if the sum of elements from i -th to n -th equals $S/3$ and 0 — otherwise. Now, to calculate the answer we have to find the sum $cnt[j] + cnt[j+1] + \dots + cnt[n]$ faster than $O(n)$. There are a lot of required ways to do this, but the easiest one is to create a new additional array $sums[]$ where in j -th element will be $cnt[j] + cnt[j+1] + \dots + cnt[n]$. It is easy to calculate in such way: $sums[n] = cnt[n]$, $sums[i] = sums[i+1] + cnt[i]$ ($i < n$).

Thus, we receive very simple solution: for each prefix of initial array $1..i$ with the sum that equals $S/3$ we need to add to the answer $sums[i+2]$.

Complexity: $O(n)$

```
n = int(input())
a = [int(x) for x in input().split()]
s = sum(a)

if s%3 != 0:
    print(0)
else:
    s = s//3

    ss = 0
    cnt = [0]*1000010
```

```

for i in range(n-1,-1,-1):
    ss += a[i]
    if ss == s:
        cnt[i] = 1

for i in range(n-2,-1,-1):
    cnt[i] += cnt[i+1]

ans = 0
ss = 0
for i in range(n-2):
    ss += a[i]
    if ss == s:
        ans += cnt[i+2]

print(ans)

```

455A. Boredom

dp, 1500, <https://codeforces.com/contest/455/problem/A>

Alex doesn't like boredom. That's why whenever he gets bored, he comes up with games. One long winter evening he came up with a game and decided to play it.

Given a sequence a consisting of n integers. The player can make several steps. In a single step he can choose an element of the sequence (let's denote it a_k) and delete it, at that all elements equal to $a_k + 1$ and $a_k - 1$ also must be deleted from the sequence. That step brings a_k points to the player.

Alex is a perfectionist, so he decided to get as many points as possible. Help him.

Input

The first line contains integer n ($1 \leq n \leq 105$) that shows how many numbers are in Alex's sequence.

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^5$).

Output

Print a single integer — the maximum number of points that Alex can earn.

2020fall-cs101, 王康安, 思路: 预处理做个桶。状态转移方程: $dp[i] = \max(dp[i-1], dp[i-2] + cnt[i] * i)$ 。

```

n = int(input())
arr = list(map(int, input().split()))
dp = [0]*(max(arr) + 1)
cnt = [0]*(max(arr) + 1)
for each in arr:
    cnt[each] += 1

dp[0] = 0
dp[1] = cnt[1]
for i in range(2, max(arr)+1):
    dp[i] = max(dp[i-1], dp[i-2] + cnt[i]*i)

print(max(dp))

```

2020fall-cs101, 李思哲, 思路: 先将相同的数字打包求和, 然后在有数的地方从前往后爬, 每次只需要考虑是放弃这次还是上一次的数更好即可。这题虽然不难但是确实让我更加明白了动态规划是个怎样的过程。

2020fall-cs101, 李宗远, 思路: 以前没用过直接爆开一个大数的方法, 然后无脑递归的方法.....我听说Python是脚本语言之后一直以为这种方法是C++的专利。学到了学到了, 真好用233

```

input()
c = [0]*100001
for m in map(int, input().split()):
    c[m] += 1

dp = [0]*100001
for i in range(1, 100001):
    dp[i] = max(dp[i-1], dp[i-2] + i* c[i] )

print(max(dp))

```

```

# ref: http://codeforces.com/blog/entry/13336
# maximize the sum of numbers that we took. Let precalc array cnt.
# cnt[x] - number of integers x in array a. Now we can easily calculate the DP:
# f(i) = max( f(i-1), f(i-2) + cnt[i]*i), 2<=i<=n
# f(1) = cnt[1]
# f(0) = 0
# The answer is f(n). Asymptotics - O(n)

n = int(input())
a = [int(i) for i in input().split()]

max_value = max(a)
#print(max_value)

cnt = (max_value+1)*[0]
for i in range(n):
    cnt[a[i]] += 1

f = (max_value+1)*[0]
f[0] = 0

```

```
f[1] = cnt[1]

for i in range(2,max_value+1):
    if f[i-1] > f[i-2] + cnt[i]*i :
        f[i] = f[i-1]
    else:
        f[i] = f[i-2] + cnt[i]*i

print(f[max_value])
```

[Python a, b = b, a + b - Stack Overflow](#)

```
n = input()
s=[0]*100002
for i in map(int, input().split()):
    s[i] += i

a=b=0
for d in s:
    a,b = max(a,b),a+d

print(a)
```

545C. Woodcutters

dp/greedy, 1500, <https://codeforces.com/problemset/problem/545/C>

Little Susie listens to fairy tales before bed every day. Today's fairy tale was about wood cutters and the little girl immediately started imagining the choppers cutting wood. She imagined the situation that is described below.

There are n trees located along the road at points with coordinates x_1, x_2, \dots, x_n . Each tree has its height h_i . Woodcutters can cut down a tree and fell it to the left or to the right. After that it occupies one of the segments $[x_i - h_i, x_i]$ or $[x_i, x_i + h_i]$. The tree that is not cut down occupies a single point with coordinate x_i . Woodcutters can fell a tree if the segment to be occupied by the fallen tree doesn't contain any occupied point. The woodcutters want to process as many trees as possible, so Susie wonders, what is the maximum number of trees to fell.

Input

The first line contains integer n ($1 \leq n \leq 10^5$) — the number of trees.

Next n lines contain pairs of integers x_i, h_i ($1 \leq x_i, h_i \leq 10^9$) — the coordinate and the height of the i -th tree.

The pairs are given in the order of ascending x_i . No two trees are located at the point with the same coordinate.

Output

Print a single number — the maximum number of trees that you can cut down by the given rules.

因为要让更多的树被砍到，而一棵树是否被砍倒只与临近的两颗树相关，所以能倒就倒，不能往左就往右，所以直接暴力判断了。（向右倒会占用下一棵树左边的空间，所以要比向左边倒的情况多考虑一点！）

```
n = int(input())
s = [[int(x) for x in input().split()] for i in range(n)]
count = 2
if n == 1:
    print(1)
else:
    for i in range(1, n-1):
        if s[i][0] - s[i-1][0] > s[i][1]:
            count += 1
        elif s[i+1][0] - s[i][0] > s[i][1]:
            count += 1
            s[i][0] += s[i][1]

    print(count)
```

2020fall-cs101-黄旭

解题思路：先首尾两棵树肯定是可以砍的，接下来中间的树肯定是能砍就尽量砍，然后记录砍树之后的被占用最大值即可。

```
x = int(input())
lt = []
n = 0
for i in range(x):
    lt.append([int(i) for i in input().split()])

for i in range(1,x-1):
    a, b = lt[i]
    if a - b > lt[i-1][0]:
        n += 1
    elif a+b < lt[i+1][0]:
        n += 1
        lt[i][0] = a + b
print([n+1,n+2][x!=1])
```

```
n = int(input())

xh_pair = []
for i in range(n):
    x,h = map(int,input().split())
    xh_pair.append((x,h))

nCount = 1
pre = xh_pair[0][0]
for index in range(1,n-1):
    i = xh_pair[index]
    j = xh_pair[index+1]
    if i[0]-pre>i[1]:
```

```

        nCount += 1
        pre = i[0]
        #print(i)
        continue
    if i[0]+i[1]<j[0]:
        nCount += 1
        #print(i)
        pre = i[0]+i[1]
        continue

pre = i[0]

if n==1:
    print(1)
else:
    print(nCount+1)

```

1000B. Light It Up

greedy, 1500, <https://codeforces.com/problemset/problem/1000/B>

Recently, you bought a brand new smart lamp with programming features. At first, you set up a schedule to the lamp. Every day it will turn power on at moment 0 and turn power off at moment M . Moreover, the lamp allows you to set a program of switching its state (states are "lights on" and "lights off"). Unfortunately, some program is already installed into the lamp.

The lamp allows only *good* programs. Good program can be represented as a non-empty array a , where $0 < a_1 < a_2 < \dots < a_{|a|} < M$. All a_i must be integers. Of course, preinstalled program is a good program.

The lamp follows program a in next manner: at moment 0 turns power and light on. Then at moment a_i the lamp flips its state to opposite (if it was lit, it turns off, and vice versa). The state of the lamp flips instantly: for example, if you turn the light off at moment 1 and then do nothing, the total time when the lamp is lit will be 1. Finally, at moment M the lamp is turning its power off regardless of its state.

Since you are not among those people who read instructions, and you don't understand the language it's written in, you realize (after some testing) the only possible way to alter the preinstalled program. You can **insert at most one** element into the program a , so it still should be a *good* program after alteration. Insertion can be done between any pair of consecutive elements of a , or even at the beginning or at the end of a .

Find such a way to alter the program that the total time when the lamp is lit is maximum possible. Maybe you should leave program untouched. If the lamp is lit from x till moment y , then its lit for $y-x$ units of time. Segments of time when the lamp is lit are summed up.

Input

First line contains two space separated integers n and M ($1 \leq n \leq 10^5$, $2 \leq M \leq 10^9$) — the length of program a and the moment when power turns off.

Second line contains n space separated integers a_1, a_2, \dots, a_n ($0 < a_1 < a_2 < \dots < a_n < M$) — initially installed program a .

Output

Print the only integer — maximum possible total time when the lamp is lit.

```
# ref: https://blog.csdn.net/tianwei0822/article/details/80884065
# 分析：加一次操作相当于把某个数字拆成两个，肯定为一开一关，让改变的这个区间长度开灯时间尽可能长，
# 即让开最大（即该数字-1）
# 操作之后的开灯时长为：该数字之前的总开灯时长+该数字之后的总关灯时长+本区间改变之后的开灯时长。
#
# 记录到第i次操作时亮灯时长，记为b[i]。
#
# 如本区间长度大于1且为开灯状态，操作后开灯时长为：b[i] + m-a[i]-(b[n+1]-b[i]) - 1,
# 如本区间长度大于1且为关灯状态，操作后开灯时长为：b[i] + m-a[i]-(b[n+1]-b[i]) + a[i]-a[i-1]-1
# 所以遍历一遍维护最大值即可。另外可以不操作，此时开灯时长为b[n+1]。

f = 1 #switch
n, M = map(int, input().split())
a = [0] + [int(x) for x in input().split()] + [M]

b = [0]*(n+2)
for i in range(1,n+1):
    b[i] = b[i-1] + f*(a[i]-a[i-1])
    f ^= 1 #0->1 or 1->0

b[n+1] = b[n] + f*(M-a[n])

ans = b[n+1] #untouched

for i in range(1,n+2):
    if (a[i]-a[i-1]>1):
        if i&1:
            ans = max(ans, b[i]+M-a[i]-(b[n+1]-b[i])-1)
        else:
            ans = max(ans, b[i]+a[i]-a[i-1]-1+M-a[i]-(b[n+1]-b[i]))

print(ans)
```

CF 1000B. greedy, 1500, <https://codeforces.com/problemset/problem/1000/B>

2020fall-cs101-顾臻宜，赵春源，解题思路和源码：

$a_k, 1 \leq k \leq n; \quad b_k = a_k - a_{k-1}, 1 \leq k \leq n+1, a_0=0, a_{n+1}=M;$ 若不额外添加开关： $ans = \sum b_{2i-1}$

若额外添加开关：由题意，开灯的时间越长越好，所以就算额外添加开关，也必须在一次关灯 a_i 的前或后1个时间单位添加。

计算ans：添加处及之前的奇数项，加上添加处之后的（本来的）偶数项，再减去添加处与本来开灯的时间点相差的1个时间单位。

先计算出插入新操作的答案，再考虑怎么插有可能更新答案。

在位置 x 处插入的本质是使 x 后开着的时间变成关着的，关着的时间变成开着的。

经过画图思考知道在一次关灯的前后1个时间单位添加，是等效的，所以只考虑关灯后的下一个时刻（即关灯后的下一时刻是最优的，不会劣于其他情况）。我们枚举位置，计算答案取max即可。

```
n, m = map(int, input().split())
a = [0] + [int(x) for x in input().split()] + [m]
tot = ans = s = 0
for i in range(1, len(a), 2):
    tot += a[i] - a[i-1]
ans = tot #总开灯时间
for i in range(2, len(a), 2):
    s += a[i-1] - a[i-2] #前 i 次开灯时间
    if a[i] > a[i-1] + 1: #若关灯时长大于 1
        t = tot - s #导出后 n-i 次开灯时间
        ans = max(ans, s + m - a[i-1] - t - 1) #之前奇数项+之后偶数项-1
print(ans)
```

134B. Multiply by 2, divide by 6

math, 900, <http://codeforces.com/problemset/problem/1374/B>

You are given an integer n . In one move, you can either multiply n by two or divide n by 6 (if it is divisible by 6 without the remainder).

Your task is to find the minimum number of moves needed to obtain 1 from n or determine if it's impossible to do that.

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \leq t \leq 2 \cdot 10^4$) — the number of test cases. Then t test cases follow.

The only line of the test case contains one integer n ($1 \leq n \leq 10^9$).

Output

For each test case, print the answer — the minimum number of moves needed to obtain 1 from n if it's possible to do that or -1 if it's impossible to obtain 1 from n .

```
for _ in range(int(input())):
    t = int(input())
    cnt = 0
    while t != 1:
        if t % 3 == 0 and t % 2 == 0:
            t = t // 6
            cnt += 1
        elif t % 3 == 0 and t % 2 != 0:
            t *= 2
            cnt += 1
        else:
            print(-1); break
    else:
        print(cnt)
```

2020fall-cs101-顾臻宜的解题思路：只要 $n=2^x3^y$ 且 $y \geq x, x, y \in \mathbb{N}$ 即可，且最多进行 $2\log_3 n$ 步。

小知识：import math之后math.log（真数N，底数a）就是 $\log_a N$ 。

```
import math

for _ in range(int(input())):
    n = int(input())
    x = 0
    for i in range(2*(1+int(math.log(n,3)))):
        if n/6 == int(n/6):
            n /= 6
            x += 1
        else:
            if n/3 == int(n/3):
                n *= 2
                x += 1

    print(x if n==1 else -1)
```

2020fall-cs101-黄旭思路：如果一个数在经过题目所说操作之后可以得到 1，那么一定在每一个步骤中都是 3 的倍数，于是在循环中一旦发现该数不是 3 的倍数，就跳出循环，反之就一直进行*2或者/6的操作，直到等于一为止，记录操作次数，若结果为 1 则输出操作次数，反之输出-1。

```
for i in range(int(input())):
    x = int(input())
    a = 0
    while x!=1:
        if x%3:
            break
        elif x%6==0:
            x=x/6
            a+=1
        else:
            x*=2
            a+=1
    print([a,-1][x!=1])
```

230B. T-primes

binary search/implementation/math/number theory, 1300, <http://codeforces.com/problemset/problem/230/B>

We know that prime numbers are positive integers that have exactly two distinct positive divisors. Similarly, we'll call a positive integer t T-prime, if t has exactly three distinct positive divisors.

You are given an array of n positive integers. For each of them determine whether it is T-prime or not.

Input

The first line contains a single positive integer, n ($1 \leq n \leq 10^5$), showing how many numbers are in the array. The next line contains n space-separated integers x_i ($1 \leq x_i \leq 10^{12}$).

Please, do not use the %lld specifier to read or write 64-bit integers in C++. It is advised to use the cin, cout streams or the %I64d specifier.

Output

Print n lines: the i -th line should contain "YES" (without the quotes), if number x_i is T-prime, and "NO" (without the quotes), if it isn't.

数论是有趣和优美的数学分支。欧几里得对于素数无穷性的证明在今天看来仍和两千年前一样清晰和优雅。长久以来，计算机都被用来辅助数论研究，有很多精妙的算法能够帮上忙。

求解素数的三种方法，包括：试除法（trial division）、埃氏筛（Sieve of Eratosthenes）、欧拉筛（Sieve of Euler，线性法），<https://blog.dotcpp.com/a/69737>

数据类型时间复杂度，<https://wiki.python.org/moin/TimeComplexity>

埃氏筛法，时间复杂度为： $O(n \cdot \log n)$ 。Python3, Accepted, 1154ms

```
# http://codeforces.com/problemset/problem/230/B

# https://www.geeksforgeeks.org/python-program-for-sieve-of-eratosthenes/
# Python program to print all primes smaller than or equal to
# n using Sieve of Eratosthenes

def SieveOfEratosthenes(n, prime):
    p = 2
    while (p * p <= n):

        # If prime[p] is not changed, then it is a prime
        if (prime[p] == True):

            # Update all multiples of p
            for i in range(p * 2, n+1, p):
                prime[i] = False
            p += 1

    n = int(input())
    x = [int(i) for i in input().split()]

    s = [True]*(10**6+1)

    SieveOfEratosthenes(10**6, s)

    for i in x:
        if i<4:
            print('NO')
            continue
        elif int(i**0.5)**2 != i:
            print('NO')
            continue
        print(['NO', 'YES'][s[int(i**0.5)]])
        #if s[int(i**0.5)]:
        #    print('YES')
        #else:
        #    print('NO')
```

埃氏筛法，时间复杂度为： $O(n \cdot \log \log n)$ 。Python3, Accepted, 1558ms

这里有一个小优化，j 从 $i * i$ 而不是从 $i + i$ 开始，因为 $i * (2 \sim i-1)$ 在 $2 \sim i-1$ 时都已经被筛去，所以从 $i * i$ 开始。

According to [Python wiki: Time complexity](#), **set** is implemented as a [hash table](#). So you can expect to lookup/insert/delete in **$O(1)$** average. <https://stackoverflow.com/questions/7351459/time-complexity-of-python-set-operations>

```
n = 1000000
a = [1] * n
s = set()

#directly add the square of prime into a set, then check if num_input is in set.
for i in range(2,n):
    if a[i]:
        s.add(i*i)
        for j in range(i*i,n,i):
            a[j] = 0

input()
for x in map(int,input().split()):
    print(["NO", "YES"][x in s])
```

埃氏筛法，by 2020fall-cs101, 汪元正, Python3, Accepted, 1340ms

```
a = [1]*(10**6)
a[0] = 0
for i in range(1,10**3,1):
    if a[i]==1:
        for j in range(2*i+1,10**6,i+1):
            a[j]=0

n = int(input())
l = [int(x) for x in input().split()]
for i in range(n):
    m = l[i]
    if m**0.5%1==0:
        r = int(m**0.5)
        if a[r-1]==1:
            print('YES')
        else:
            print('NO')
    else:
        print('NO')
```

线性筛（欧拉筛），时间复杂度为： $O(n)$ 。Python3, Accepted, 1808ms。

```
# https://blog.dotcpp.com/a/69737
# https://blog.csdn.net/xuechen_gemgirl/article/details/79555123
def euler(r):
    prime = [0 for i in range(r+1)]
    common = []
```

```

for i in range(2, r+1):
    if prime[i] == 0:
        common.append(i)
    for j in common:
        if i*j > r:
            break
        prime[i*j] = 1
        if i % j == 0:
            break
    return prime

s = euler(1000000)
#print(s)

input()
for i in map(int,input().split()):
    if i<4:
        print('NO')
        continue
    elif int(i**0.5)**2 != i:
        print('NO')
        continue
    if s[int(i**0.5)]==0:
        print('YES')
    else:
        print('NO')

```

270A. Fancy Fence

geometry/implementation/math, 1100, x23265, <https://codeforces.com/problemset/problem/270/A>

Emuskald needs a fence around his farm, but he is too lazy to build it himself. So he purchased a fence-building robot.

He wants the fence to be a regular polygon. The robot builds the fence along a single path, but it can only make fence corners at a single angle α .

Will the robot be able to build the fence Emuskald wants? In other words, is there a regular polygon which angles are equal to α ?

Input

The first line of input contains an integer t ($0 < t < 180$) — the number of tests. Each of the following t lines contains a single integer α ($0 < \alpha < 180$) — the angle the robot can make corners at measured in degrees.

Output

For each test, output on a single line "YES" (without quotes), if the robot can build a fence Emuskald wants, and "NO" (without quotes), if it is impossible.

2020fall-cs101-黄旭: 对于 n 边形, 其内角和为 $(n-2) \cdot 180^\circ$, 内角为 x 度, 有 $(n-2) \cdot 180 = n \cdot x$. 则可以得到 n 的表达式为 $n = 360 / (180 - x)$, 若 n 为整数, 则可行, 反之不可行。

```
for i in range(int(input())):
    x=int(input())
    print(['NO', 'YES'][360%(180-x)==0])
```

```
n=int(input())
def check(x):
    if 360%(180-x)==0:
        return"YES"
    else:
        return"NO"

for i in range(n):
    x=int(input())
    print(check(x))
```

1221A. 2048 Game

brute force/greedy/math, 1000, x11647, <http://codeforces.com/problemset/problem/1221/A>

You are playing a variation of game 2048. Initially you have a multiset s of n integers. Every integer in this multiset is a power of two.

You may perform any number (possibly, zero) operations with this multiset.

During each operation you choose two **equal** integers from s , remove them from s and insert the number equal to their sum into s .

For example, if $s=\{1,2,1,1,4,2,2\}$ and you choose integers 2 and 2, then the multiset becomes $\{1,1,1,4,4,2\}$.

You win if the number 2048 belongs to your multiset. For example, if $s=\{1024,512,512,4\}$ you can win as follows: choose 512 and 512, your multiset turns into $\{1024,1024,4\}$. Then choose 1024 and 1024, your multiset turns into $\{2048,4\}$ and you win.

You have to determine if you can win this game.

You have to answer q independent queries.

Input

The first line contains one integer q ($1 \leq q \leq 100$) – the number of queries.

The first line of each query contains one integer n ($1 \leq n \leq 100$) — the number of elements in multiset.

The second line of each query contains n integers s_1, s_2, \dots, s_n ($1 \leq s_i \leq 2^{29}$) — the description of the multiset. It is guaranteed that all elements of the multiset are powers of two.

Output

For each query print YES if it is possible to obtain the number 2048 in your multiset, and NO otherwise.

You may print every letter in any case you want (so, for example, the strings yEs, yes, Yes and YES will all be recognized as positive answer).

```
# SHEN Tianfang, 2020/10/13
q=int(input())
l=[1,2,4,8,16,32,64,128,256,512,1024]
for i in range(q):
    n=int(input())
    s=[int(x) for x in input().split()]

    for i in range(11):
        while s.count(l[i])>1:
            s.remove(l[i])
            s.remove(l[i])
            s.append(2*l[i])

    if 2048 in s:
        print('YES')
    else:
        print('NO')
```

```
for t in range(int(input())):
    n = input()
    l = filter(lambda x : x <= 2048, map(int, input().split()) )
    print('YES' if sum(l) >= 2048 else 'NO')
```

2020fall-cs101-邹京驰, 版本 1: 学习了列表型对象的 count方法。按照 2048原本玩法的逻辑顺序, 对于先天没有 2048的情况, 先把全部 1全部合成为 2 (可能剩下单个 1无法合成, 则略), 再把 2 (包括上一步用 1合成的) 全部合成为 4, 以此类推, 直到把 1024全部合成为 2048, 若合成出的 2048的数量大于 1则 YES, 否则NO.

```
q = int(input())
for i in range(0, q):
    n = int(input())
    s = list(map(int, input().split()))
    if 2048 in s:
        print('YES')
    else:
        t = 0
        for j in range(0, 11):
            t1 = s.count(2**j) + t
            t = t1//2
        if t > 0:
            print('YES')
        else:
            print('NO')
```

2020fall-cs101-邹京驰, 版本 2: 注意到数学: 只需把小于等于 2048的元素加和大于等于 2048即满足条件。原因: 在 2048一下因为落单而被舍弃的元素达到最多时加和为 $1+2+4+8+\dots+512+1024=2047$, 这一加和本来就不到 2048, 且新增1~1024的任何一个元素就可以生成 2048.因此生成新的 2048当且仅当1~1024的元素直接加和达到 2048。再并入先天有 2048的情况, 只需把小于等于 2048的元素加和大于等于 2048即满足条件。


```

q = int(input())
for i in range(0, q):
    n = int(input())
    s = list(map(int, input().split()))
    su = 0
    for j in range(0, n):
        if s[j] <= 2048:
            su = su + s[j]

    if su >= 2048:
        print('YES')
    else:
        print('NO')

```

1427B. Chess Cheater

greedy/implementation/sortings, 1400, x5683, <https://codeforces.com/problemset/problem/1427/B>

You like playing chess tournaments online.

In your last tournament you played n games. For the sake of this problem, each chess game is either won or lost (no draws). When you lose a game you get 0 points. When you win you get 1 or 2 points: if you have won also the previous game you get 2 points, otherwise you get 1 point. If you win the very first game of the tournament you get 1 point (since there is not a "previous game").

The outcomes of the n games are represented by a string s of length n : the i -th character of s is W if you have won the i -th game, while it is L if you have lost the i -th game.

After the tournament, you notice a bug on the website that allows you to change the outcome of **at most** k of your games (meaning that at most k times you can change some symbol L to W, or W to L). Since your only goal is to improve your chess rating, you decide to cheat and use the bug.

Compute the maximum score you can get by cheating in the optimal way.

Input

Each test contains multiple test cases. The first line contains an integer t ($1 \leq t \leq 20,000$) — the number of test cases. The description of the test cases follows.

The first line of each testcase contains two integers n, k ($1 \leq n \leq 100,000$, $0 \leq k \leq n$) – the number of games played and the number of outcomes that you can change.

The second line contains a string s of length n containing only the characters W and L. If you have won the i -th game then $s_i = W$, if you have lost the i -th game then $s_i = L$.

It is guaranteed that the sum of n over all testcases does not exceed 200,000.

Output

For each testcase, print a single integer – the maximum score you can get by cheating in the optimal way.

思路: <https://www.bbsmax.com/A/WpdKEVpmjV/>

请注意, 分数等于 $[\text{score} = 2 \cdot \#\{\text{wins}\} - \#\{\text{winning_streaks}\}]$, 连胜是连续获胜的最大顺序。

在下面的说明中，变量($\# \{wins\}$, $\# \{winning_streaks\}$) 始终与初始情况相关。

如果 $(k + \# \{wins\} \geq n)$ ，则有可能赢得所有比赛，因此答案为 $(2n-1)$ 。

否则，很明显，我们要转换 k 获胜中的 k 损失。因此，作弊后，获胜次数将为 $(k + \# \{wins\})$ 。考虑到以上公式，仍然仅是要减少获胜间隔的数量。

我们如何才能减少获胜间隔的数量？非常直观的是，我们将从长度最短的差距开始，以“填补”连续的获胜间隔之间的差距。可以证明，如果没有填补 g 个缺口（即在作弊之后， g 个缺口仍然至少包含一个损失），则至少有 $g + 1$ 个获胜间隔。

实现过程如下。通过线性扫描，我们可以找到间隙的长度，然后对它们进行排序。最后，我们计算可以选择的总和 ($\leq k$) 的数量。答案是

$$[2 \cdot (k + \# \{wins\}) - \# \{winning_streaks\} + \# \{gaps_we_can_fill\}]$$

解决方案的复杂度为 $(O(\log(n)))$ 。

```
#include<bits/stdc++.h>
#define ms(a,b) memset(a,b);
using namespace std;
typedef long long ll;
int main() {
    //freopen("in.txt", "r", stdin);
    ios_base::sync_with_stdio(false), cin.tie(0), cout.tie(0);
    int T;
    cin >> T;
    for (int t = 1; t <= T; t++) {
        int N, K;
        cin >> N >> K;
        string S;
        cin >> S;
        int winning_streaks_cnt = 0;
        int wins = 0;
        int losses = 0;
        vector<int> losing_streaks;
        for (int i = 0; i < N; i++) {
            if (S[i] == 'W') {
                wins++;
                if (i == 0 or S[i - 1] == 'L') winning_streaks_cnt++;
            }
            if (S[i] == 'L') {
                losses++;
                if (i == 0 or S[i - 1] == 'W') losing_streaks.push_back(0);
                losing_streaks.back()++;
            }
        }
        if (K >= losses) {
            cout << 2 * N - 1 << "\n";
            continue;
        }
        if (wins == 0) {
            if (K == 0) cout << 0 << "\n";
            else cout << 2 * K - 1 << "\n";
            continue;
        }
        if (S[0] == 'L') losing_streaks[0] = 1e8;
```

```

        if (S[N - 1] == 'L') losing_streaks.back() = 1e8;
        sort(losing_streaks.begin(), losing_streaks.end());
        wins += K;
        for (int ls : losing_streaks) {
            if (ls > K) break;
            K -= ls;
            winning_streaks_cnt--;
        }
        cout << 2 * wins - winning_streaks_cnt << "\n";
    }
}

```

```

# ref: https://www.bbsmax.com/A/wpdKEVpmJV/
for _ in range(int(input())):
    N,K = map(int, input().split())
    S = input()
    winning_streaks_cnt = wins = losses = 0
    losing_streaks = []

    for i in range(N):
        if S[i] == 'W':
            wins += 1
            if i==0 or S[i-1]=='L':
                winning_streaks_cnt += 1

        if S[i]=='L':
            losses += 1
            if i==0 or S[i-1]=='W':
                losing_streaks.append(0)
            losing_streaks[-1] = losing_streaks[-1] + 1

    if K >= losses:
        print(2*N-1)
        continue

    if wins == 0:
        if K == 0:
            print(0)
        else:
            print(2*K-1)
        continue

    if S[0]=='L':
        losing_streaks[0] = 1e8
    if S[-1]=='L':
        losing_streaks[-1] = 1e8

    losing_streaks.sort()
    wins += K
    for ls in losing_streaks:
        if ls > K:
            break

    K -= ls
    winning_streaks_cnt -= 1

```

```
print(2*wins - winning_steaks_cnt)
```

References:

- [1]. Siru Chen, 20171226-Codeforces-top100_good-solution.pdf
- [2]. Student assignments, fall 2020.
- [3]. <https://csrgxtu.github.io/2015/03/20/Writing-Mathematic-Fomulars-in-Markdown/>