

1st Mohaghegh Collegiate
Programming Contest
MCPC



اولین دوره مسابقات برنامه نویسی
دانشجویی دانشگاه محقق اردبیلی
۱۰ آذر ۱۳۹۴





می خواهیم در یک جمله ورودی کاراکتری که بزرگترین کد اسکی را دارد پیدا کرده و چاپ نماییم. جمله ورودی می تواند شامل کلمات و خط فاصله باشد. هر کلمه تنها می تواند از حروف بزرگ یا کوچک انگلیسی تشکیل شده باشد. انتهای جمله با کلید Enter مشخص می گردد.

Mohagheh Collegiate Programming Contest	نمونه ورودی:
t	نمونه خروجی:



علی که برنامه نویسی را به تازگی یاد گرفته است علاقه زیادی به اعداد دارد. او میخواهد در دنباله ای از اعداد ببیند که کدام عدد تکرار شده است. اما خوشبختانه علی یک ویژگی مهم درباره این دنباله را هم میداند. و آن این است که اگر مثلاً دنباله شامل n عدد باشد، تمامی اعداد بین ۱ تا $n-1$ هستند و فقط یک عدد در این دنباله تکرار شده است. شما باید برنامه‌ای بنویسید که به علی در پیدا کردن این عدد کمک کند.

ورودی:

شامل تعدادی سناریو است (حداکثر 100 سناریو) که هر سناریو در خط اول شامل عدد n ($1 < n \leq 1000$) و در خط بعدی شامل n عدد است که هر کدام در بازه ی ۱ تا $n-1$ قرار دارند. تضمین میشود که تنها یک عدد در این دنباله تکراری است. توجه کنید.

خروجی:

در هر خط عدد تکراری را چاپ کنید.

نمونه خروجی	نمونه ورودی
4	5 1 2 3 4 4
1	2 1 1
3	4 3 1 2 3
3	10 1 7 3 8 6 5 4 9 2 3



دو دنباله از اعداد صحیح به نام های X و Y داریم که تعداد عناصرشان یکسان است اما ترتیب خاصی ندارند. ضرب دنباله ها را بصورت ضرب عناصر متناظر دو آرایه با اندیس های یکسان و جمع کردن نتایج حاصل تعریف می کنیم. برای مثال برای دو دنباله زیر :

$$X=2, 4, 3, 5$$

$$Y=31, 7, 9, 2$$

$$\text{Result}= 2*31+4*7+3*9+5*2 =127$$

می خواهیم عناصر دو دنباله را به گونه ای مرتب کنیم که حاصل ضرب این دنباله ها حداقل شود.

ورودی:

خط اول ورودی حاوی یک عدد صحیح N ($1 \leq N \leq 1000$) که نشان دهنده تعداد عناصر هر دنباله است.

دو خط بعدی ورودی حاوی دنباله های X و Y است که همه عناصر آن بین ۱ و ۱۰۰ است.

خروجی:

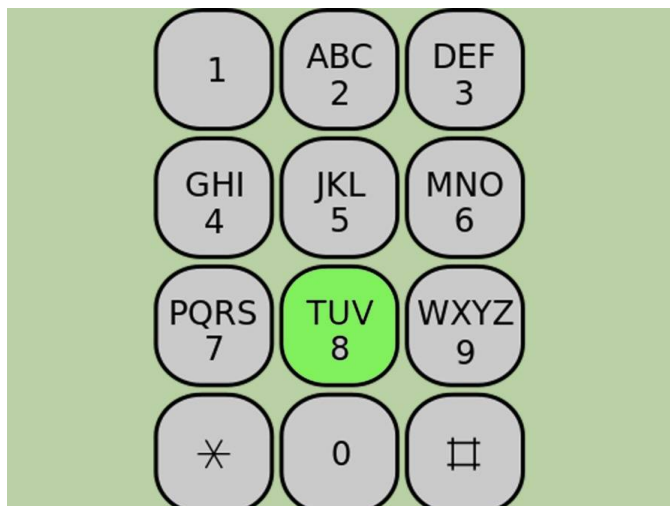
عددی چاپ شود که نشان دهنده حداقل مقداری است که برای ضرب دو دنباله ورودی می تواند بدست آید.

نمونه ورودی:	نمونه ورودی:
5 2 2 3 4 5 4 5 2 2 3	3 1 2 1 2 1 1
نمونه خروجی:	نمونه خروجی:
45	5



امروزه ارسال پیامک یکی از متداول ترین روش های ارتباط بین افراد است. با افزایش تعداد ارسال پیامک توسط افراد، نیاز به تایپ سریعتر متن پیامک بیشتر و بیشتر احساس می شود.

فرض می کنیم که هر پیام، از رشته ای از حروف انگلیسی بزرگ و خط فاصله تشکیل شده است. مطابق شکل زیر حروف 'A' تا 'Z' به کلیدهای '2' تا '9' صفحه کلید گوشی موبایل منتسب می شود. بسته به این که آن حرف چندمین حرف روی یک کلید (از چپ به راست) باشد برای تایپ کردن یک حرف بایستی کلید مربوط به آن حرف را ۱، ۲، ۳، یا ۴ بار فشار داد.



اگر دو حرف متوالی یک پیام روی یک کلید باشند، بایستی کمی منتظر ماند تا حرف قبلی روی صفحه نمایش تثبیت شود و سپس کلید بتواند برای تایپ حرف بعدی استفاده شود. برای مثال، برای تایپ حرف 'X' بایستی کلید '9' دو بار فشار داده شود. اگر حرف بعدی پیام روی همان کلید نباشد می توان ادامه پیام را بلافاصله تایپ نمود. در غیر اینصورت شخص باید زمان کوتاهی منتظر بماند تا حرف 'X' تایپ شده تثبیت شود و سپس حرف بعدی ('W'، 'X'، 'Y' یا 'Z') بتواند تایپ شود. برای تایپ خط فاصله، از کلید '1' استفاده می شود. با توجه به اینکه هیچ حرفی برای کلید '1' در نظر نگرفته شده، برای تثبیت شدن خط فاصله نیاز به زمان نمی باشد.

زمان مورد نیاز برای فشار دادن هر کلید و زمان انتظار برای تثبیت یک حرف داده شده است. برنامه شما بایستی حداقل زمان مورد نیاز برای تایپ یک رشته ناتهی را با قوانین ذکر شده بیابد.



ورودی

اولین خط ورودی حاوی اعداد p و w ($1 \leq p, w \leq 1000$) است که به ترتیب نشان دهنده مقدار زمان به میلی ثانیه برای فشار دادن یک حرف و انتظار برای تثبیت آن است. خط دوم ورودی حاوی یک رشته ناتهی با طول حداکثر ۱۰۰ است که شامل حروف بزرگ انگلیسی یا خط فاصله است. در ابتدا یا انتهای خط ورودی خط فاصله وجود ندارد.

خروجی

برای هر مورد، خروجی بایستی عددی باشد که نشان دهنده زمان مورد نیاز برای تایپ پیام بر حسب میلی ثانیه است.

<p>نمونه ورودی:</p> <p>2 10 ABBAS SALAM</p>	<p>نمونه خروجی:</p> <p>72</p>
--------------------------------------------------------	-------------------------------



برنامه ای بنویسید که مشخص کند آیا یک کلمه پالیندروم است یا خیر. یک پالیندروم دنباله ای از کاراکترها است که از هر دو طرف به یک شکل خوانده می شود (از ابتدا به انتها و انتها به ابتدا). برای مثال رشته های زیر پالیندروم هستند:

ABCCBA

A

AMA

و رشته های زیر پالیندروم نیستند:

HELLO

ABAB

PPA

ورودی

هر خط ورودی شامل حداقل ۱ و حداکثر ۸۰ کاراکتر است. فرض بر این است که ورودی فقط شامل حروف بزرگ انگلیسی است و هیچ حرف کوچک انگلیسی، علائم نشانه گذاری، ارقام یا خط فاصله در کلمه وجود ندارد.

خروجی

خروجی برنامه یکی از دو کلمه YES یا NO است که به ترتیب نشان دهنده پالیندروم بودن یا نبودن کلمه ورودی است.

نمونه خروجی:	نمونه ورودی:
YES	ABCCBA
YES	A
NO	HELLO
NO	ABAB
YES	AMA
NO	PPA



After years as a brick-layer, you've been called upon to analyze the instability of brick walls. The instability of a wall can be approximated by the maximum damage to a wall in case of taking one brick out. A brick will fall if all bricks that are directly underneath it are removed. Note that if the space underneath a brick is partially empty, it does not fall. You are given the description of all bricks in a wall, and must determine the instability of the wall as described in the following sections.

Input

There are multiple test cases in the input. Each test case consists of a single line, " $M N$ " ($1 \leq M, N \leq 100$) where M and N indicate the height and width (in units), respectively, of the input wall.

Each of the next M lines is a string of N digits which specifies a row in the wall. Each brick in a row is represented by a substring of the row (like s) such that every digit in s is the same, which is equal to the length of s too. For example, 333 and 22 are two bricks of length 3 and 2 respectively, but 111 specifies three bricks of length one. A 0 in a row means there is no brick in that place of wall. Note that the height of each brick is one. You may assume that the input is correct. This means:

1. There is no brick such that the length of the brick does not conform to the digits in the brick (like 222 in the row 12221).
2. No brick can fall initially.

Output

For each test case, write a single line containing maximum sum of the bricks' lengths that will fall if we take one brick out (including that brick).



Sample Input	Sample Output
4 5 33322 22333 33322 22333	5
4 6 122333 444422 111111 333333	8
3 3 022 220 111	4



Sometimes when Armin has nothing to do he starts thinking of new operations on strings (a sequence of characters), and also sometimes if don't say all of times these operations don't make any sense! This time Armin has invented a new operation called multiplication of strings and he defines it this way, if S and T are two strings:

"I start from the first character of S and go through all characters in order, each time I saw a character of S I will write it and then write the string T immediately after it. I'll do this until I get to the end of string S."

But unfortunately, after inventing this wonderful operation! Armin has had lunch. And when he eats lunch he feels very tired and doesn't have enough energy to write the result of this operation for a few strings! That's why you should do him a favor by writing a program to do that.

Input

Input consists of two strings of alphanumeric characters separated by a single space in each line.

Output

Print a single line that is the result of the multiplication as explained above.

Sample Input	Sample Output
Moj aRmiN	MaRmiNoaRmiNjaRmiN
12a 11	111211a11
Aaa li	Alialiali



According to Wikipedia, an arithmetic progression (AP) is a sequence of numbers such that the difference of any two successive members of the sequence is a constant. For instance, the sequence 3, 5, 7, 9, 11, 13, . . . is an arithmetic progression with common difference 2. For this problem, we will limit ourselves to arithmetic progression whose common difference is a non-zero integer.

On the other hand, a geometric progression (GP) is a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed non-zero number called the common ratio. For example, the sequence 2, 6, 18, 54, . . . is a geometric progression with common ratio 3. For this problem, we will limit ourselves to geometric progression whose common ratio is a non-zero integer.

Given three successive members of a sequence, you need to determine the type of the progression and the next successive member.

Input Format

Your program will be tested on one or more test cases. Each case is specified on a single line with three integers ($-10,000 < a_1, a_2, a_3 < 10,000$) where a_1, a_2 , and a_3 are distinct.

Output Format

For each test case, your program must print a single line of the form:

XX_V where XX is either AP or GP depending on if the given progression is an Arithmetic or Geometric Progression. V is the next member of the given sequence. All input cases are guaranteed to be either an arithmetic or geometric progression.

Sample Input	Sample Output
4 7 10	AP 13
2 6 18	GP 54