Problem A. Regex

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Write a program to find the sequences of one upper case letter followed by lower case letters.

Input

Given a string.

Output

Print "Found a match!"if given text matches to pattern. Otherwise print "Not matched!"

Examples

standard input	standard output
aab_cbbbc	Not matched!
aab_Abbbc	Found a match!
zA	Not matched!
sdafa234d!sadfa___sdfasdf%A235234z	Not matched!

Problem B. Regex 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Write a program to match a string that contains only upper and lowercase letters, numbers, and underscores.

Input

Given a string.

Output

Print "Found a match! " if given text matches to pattern. Otherwise print "Not matched! "

Examples

standard input	standard output

`The quick brown fox`
`jumps over the lazy dog.`
	`Not matched!`

`Python_Exercises_1`	`Found a match!`

Problem C. Data compressing

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given an array of size n. You need to assign each index from 1 to u for each distinct elememt of the array, where u is the amount of different numbers in the array. The less element is, the less its index.

Input

The first line of the input contains the only integer n - size of the array.

The second line contains n integers a_i - elements of the array.

Output

Print u lines. Each line must contain the index and the number that is assigned to this index. See samples for better understanding of the output.

Examples

standard input	standard output

5
8 4 2 5 9
	1 2
	2 4
	3 5
	4 8
	5 9

5
3 5 2 5 3
	1 2
	2 3
	3 5

10 1 1 2 2 1 2 1 2 2 1	1 1 2 2

Problem D. Just Map

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given names and you need to print the names of those whose repetition is an even number. Output names have to be printed in alphabetic order

Example

standard input	standard output
Ayana Ayana Dias Dias Dias Alik Alik Alik Alik	Alik Ayana

Problem E. Cities

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Jonathan is a very curious boy. He has a list of countries and list of cities of each country. His task is for each city determining country where is located. But Jonathan doesn't know geography and needs your help.

Input

In the first line of the input given an integer n.

The next n lines given the name of country, count of cities k, and cities of of this country.

It is guaranteed that their names are unique.

In the next line given m - the number of names of cities which Jonathan asked.

The next m lines given names of cities.

Output

For each Jonathan's query - print country name, if we know in which country is located. Otherwise, print "Unknown".

Example

standard input	standard output
3 Kazakshtan 3 Kyzylorda Karaganda Uralsk USA 3 California Berkly New-York England 1 London 4 Kyzylorda New-York Atyrau Karaganda	Kazakshtan USA Unknown Kazakshtan

Problem F. String shift

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

We have a string S consisting of uppercase English letters. Additionally, an integer N will be given.

Shift each character of S by N in alphabetical order (see below), and print the resulting string.

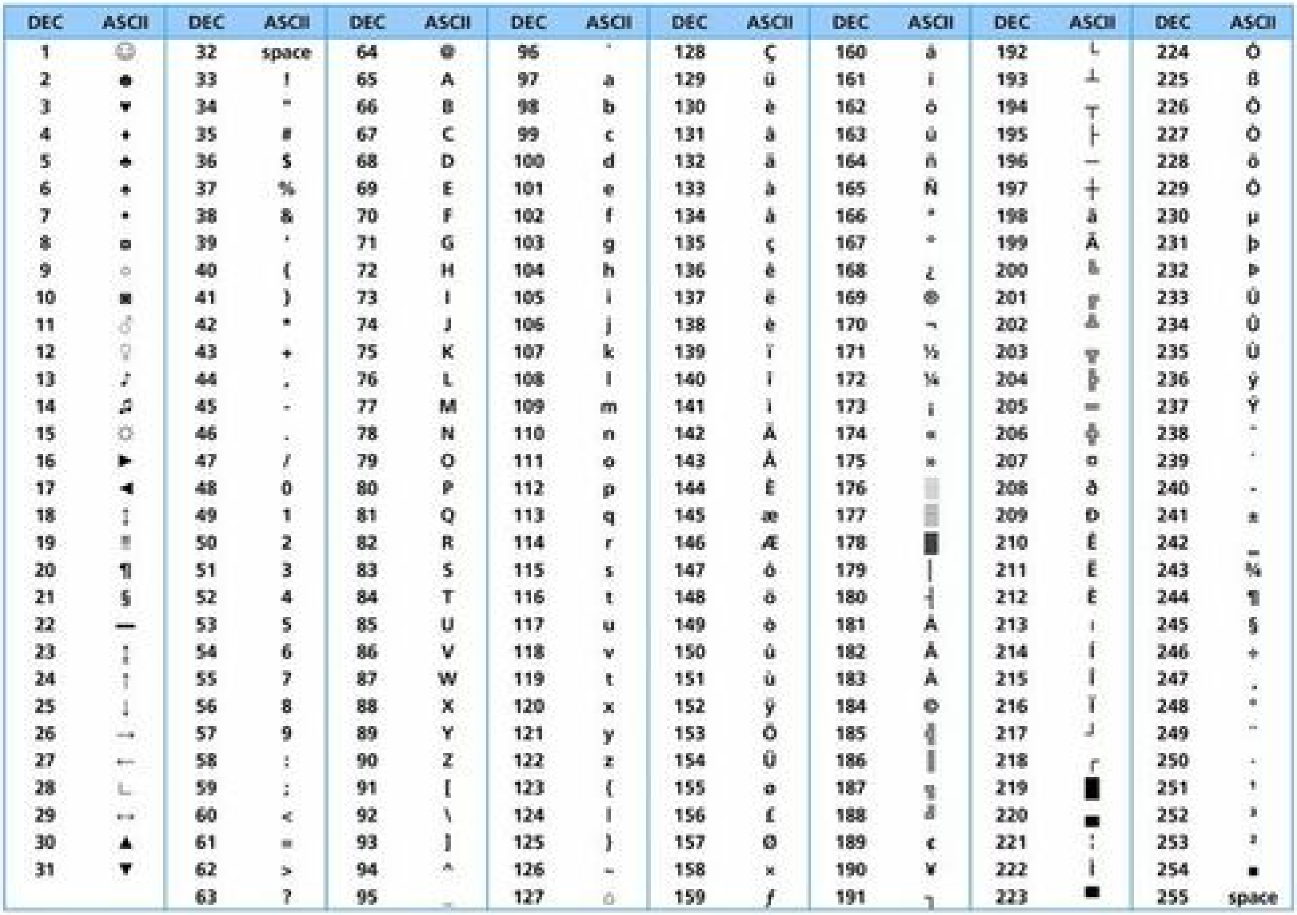
We assume that A follows Z. For example, shifting A by 2 results in C (A -> B -> C), and shifting X by 3 results in B (X -> Z -> A -> B).

$(0 \le N \le 26)$, $(1 \le |S| \le 10000)$.

Examples

standard input	standard output
2 ABCXYZ	CDEZAB
0 ABCXYZ	ABCXYZ
13 ABCDEFGHIJKLMNOPQRSTUVWXYZ	NOPQRSTUVWXYZABCDEFGHIJKLM

Note



DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII
1	☺	32	space	64	@	96	`	128	Ç	160	á	192	└	224	Ó
2	☻	33	!	65	A	97	a	129	ü	161	í	193	┴	225	ß
3	♥	34	"	66	B	98	b	130	é	162	ó	194	┬	226	Ô
4	♦	35	#	67	C	99	c	131	â	163	ú	195	├	227	Ò
5	♣	36	$	68	D	100	d	132	ä	164	ñ	196	─	228	õ
6	♠	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	Õ
7	•	38	&	70	F	102	f	134	å	166	ª	198	ã	230	µ
8	◘	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
9	○	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	Þ
10	◙	41	)	73	I	105	i	137	ë	169	®	201	╔	233	Ú
11	♂	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Û
12	♀	43	+	75	K	107	k	139	ï	171	½	203	╦	235	Ù
13	♪	44	,	76	L	108	l	140	î	172	¼	204	╠	236	ý
14	♫	45	-	77	M	109	m	141	ì	173	¡	205	═	237	Ý
15	☼	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	¯
16	►	47	/	79	O	111	o	143	Å	175	»	207	¤	239	´
17	◄	48	0	80	P	112	p	144	É	176	░	208	ð	240	-
18	↕	49	1	81	Q	113	q	145	æ	177	▒	209	Ð	241	±
19	‼	50	2	82	R	114	r	146	Æ	178	▓	210	Ê	242	‗
20	¶	51	3	83	S	115	s	147	ô	179	│	211	Ë	243	¾
21	§	52	4	84	T	116	t	148	ö	180	┤	212	È	244	¶
22	▬	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
23	↨	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
24	↑	55	7	87	W	119	t	151	ù	183	À	215	Î	247	¸
25	↓	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
26	→	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	¨
27	←	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
28	∟	59	;	91	[	123	{	155	ø	187	╗	219	█	251	¹
29	↔	60	<	92	\	124	|	156	£	188	╝	220	▄	252	³
30	▲	61	=	93	]	125	}	157	Ø	189	¢	221	¦	253	²
31	▼	62	>	94	^	126	~	158	×	190	¥	222	Ì	254	■
		63	?	95	_	127	⌂	159	ƒ	191	┐	223	▀	255	space

Problem G. Common characters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a list of strings `A`. Print all characters that appears in all strings

Input

In the first line given `n` - number of strings.

In the next `n` lines given elements of array.

Output

Print all single common characters, if there are no common characters print NO COMMON CHARACTERS

Examples

standard input	standard output
3 bella label roller	e l
4 alik diyas ali dayana	a
3 aab ab c	NO COMMON CHARACTERS

Problem H. Fence problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Adil really likes climbing over the fences, but sometimes fences are too high. He does practice 3 times a day. To climb over the fence, the average score at least of one of the training days must be more or equal than a height of the fence. You have to determine, can he climb over the given fence or not.

Input

The first line of input contains integers n ($1 \leq n \leq 1000$) - number of Adil's training days and k ($0 \leq n \leq 100$) - the height of the fence. Each of the next n lines contains 3 integers - three scores of his practice a day.

Output

Print 'YES' if he can climb over the fence otherwise print 'NO'.

Examples

standard input	standard output

3 6
1 2 3
1 1 2
5 6 7
	YES

4 12
1 2 3
4 5 6
7 8 9
12 13 14
	YES

Problem I. 75072. Interesting array

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Muratbek is fond of interesting arrays. He has an array $a_1, a_2, ..., a_n$ of n integers. Your task is to check whether his array is *interesting* or not.

An array is called *interesting* if all elements of the array are sorted in non-decreasing order. Formally, for each pair of indexes i and j, such that $1 \leq i < j \leq n$, following inequality holds: $a_i \leq a_j$.

Input

The first line of input contains a single integer n — the size of the given array ($1 \leq n \leq 100$).

The second line of input contains n space-separated integers $a_1, a_2, ..., a_n$ — the given array ($0 \leq a_i \leq 1000$).

Output

If Muratbek's array is *interesting*, print «Interesting» (without quotes).

Otherwise, print «Not interesting» (without quotes).

Examples

standard input	standard output
5 1 2 8 9 25	Interesting
4 2 5 4 3	Not interesting
1 5	Interesting

Problem J. Plates

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Shah is dating with Zhasmin. Tomorrow is the Zhasmin's birthday. Now, he needs some money. He works in a restaurant. His task is to count the number of clean and dirty plates.

Input

You are given N. N-number of elements.

Output

Print the number of each plate. "Clean: Dirty: "

Examples

standard input	standard output
5 1 1 1 1 1	Clean:0 Dirty:5
3 1 0 1	Clean:0 Dirty:3
3 1 1 1	Clean:0 Dirty:3
2 1 1	Clean:0 Dirty:2
2 0 1	Clean:0 Dirty:2
1 0	Clean:1 Dirty:0