

1 to N

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

Given a number N . Print numbers from **1** to N in separate lines.

Input

Only one line containing a number N ($1 \leq N \leq 10^3$).

Output

Print N lines according to the required above.

Example

standard input	standard output
5	1 2 3 4 5

Even Numbers

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

Given a number N . Print all **even** numbers between **1** and N inclusive in separate lines.

Input

Only one line containing a number N ($1 \leq N \leq 10^3$).

Output

Print the answer according to the required above. If there are no **even** numbers print **-1**.

Examples

standard input	standard output
10	2 4 6 8 10
5	2 4

Even, Odd, Positive and Negative

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

Given N numbers. **Count** how many of these values are **even**, **odd**, **positive** and **negative**.

Input

First line contains one number N ($1 \leq N \leq 10^3$) number of values.

Second line contains N numbers ($-10^5 \leq X_i \leq 10^5$).

Output

Print four lines with the following format:

First Line: "Even: X ", where X is the number of **even** numbers in the given input.

Second Line: "Odd: X ", where X is the number of **odd** numbers in the given input.

Third Line: "Positive: X ", where X is the number of **positive** numbers in the given input.

Fourth Line: "Negative: X ", where X is the number of **negative** numbers in the given input.

Example

standard input	standard output
5 -5 0 -3 -4 12	Even: 3 Odd: 2 Positive: 1 Negative: 3

Note

First Example :

Even Numbers are : **0, -4 , 12**

Odd Numbers are : **-5 , -3**

Positive Numbers are : **12**

Negative Numbers are : **-5 , -3 , -4**

Fixed Password

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

Given multiple lines each line contains a number X which is a password. Print “**Wrong**” if the password is **incorrect** otherwise, print “**Correct**” and **terminate** the program.

Note: The “**Correct**” password is the number **1999**.

Input

The input contains several passwords.

Each line contains a number X ($10^3 \leq X \leq 10^4 - 1$).

Output

Print “**Wrong**” if the password is typed **wrong** otherwise, print “**Correct**” if the password is typed **correctly**.

Example

standard input	standard output
2200	Wrong
1020	Wrong
1999	Correct
1000	
9999	

Max

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

Given a number N , and N numbers, find **maximum** number in these N numbers.

Input

First line contains a number N ($1 \leq N \leq 10^3$).

Second line contains N numbers X_i ($0 \leq X_i \leq 10^9$).

Output

Print the **maximum** number.

Example

standard input	standard output
5 1 8 5 7 5	8

Multiplication table

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 64 megabytes

Given a number N . Print the **multiplication table** of the number from 1 to 12

For example: if $N = 1$

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

Input

Only one line containing a number N ($1 \leq N \leq 50$).

Output

Print **12** lines according to the required above.

Examples

standard input	standard output
1	1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 1 * 4 = 4 1 * 5 = 5 1 * 6 = 6 1 * 7 = 7 1 * 8 = 8 1 * 9 = 9 1 * 10 = 10 1 * 11 = 11 1 * 12 = 12
2	2 * 1 = 2 2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18 2 * 10 = 20 2 * 11 = 22 2 * 12 = 24

Factorial

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 64 megabytes

Given a number N . Print the **factorial** of number N .

Input

First line contains a number T ($1 \leq T \leq 15$) number of test cases.

Next T lines will contain a number N ($0 \leq N \leq 20$)

Output

For each test case print a single line contains the **factorial** of N .

Example

standard input	standard output
2	120
5	6
3	

Note

Factorial, in mathematics, the product of all positive integers less than or equal to a given positive integer and denoted by that integer and an exclamation point.

Thus, factorial seven is written $7!$, meaning $1 * 2 * 3 * 4 * 5 * 6 * 7 = 5040$.

Factorial zero is defined as equal to 1.

In first test case for $N = 5$, $5! = 1 * 2 * 3 * 4 * 5 = 120$ so the answer is **120**.

In Second test case for $N = 3$, $3! = 1 * 2 * 3 = 6$ so the answer is **6**.

One Prime

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 64 megabytes

Given a number X . Determine if the number is **prime** or **not**

Note:

A **prime** number is a number that is greater than **1** and has only two factors which are **1** and **itself**.

In other words : **prime number divisible only by 1 and itself**.

Be careful that 1 is not prime .

The first few **prime** numbers are

2 3 5 7 11
19 23 29 31
43 47 53 59
71 73 79 83

Input

Only one line containing a number X ($2 \leq X \leq 10^5$).

Output

print “YES” if the number is **prime** and “NO” otherwise.

Examples

standard input	standard output
7	YES
15	NO

Note

First Example :

7 is prime because it is not divisible by **2,3,4,5,6**, and only divisible by 1 and itself, so the answer is **YES**.

Second Example :

15 not is prime because it is divisible by **3 ,5**, so the answer is **NO**.

Palindrome

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

Given a number N . Print **2** lines that contain the following respectively:

1. Print N in a reversed order and **not leading zeroes**.
2. If N is a **palindrome number** print “**YES**” otherwise, print “**NO**”.

Note:

A **palindrome number** is a number that reads the same forward or backward.

For example: 12321, 101 are **palindrome numbers**, while 1201, 221 are **not**.

A **leading zero** is any 0 digit that comes before the first nonzero digit in a number for example : numbers (005 , 01 , 0123 , 02 , 000250) are leading zeroes but (5 , 123 , 20 ,2500) not leading zeroes numbers .

Input

Only one line containing a number N ($1 \leq N \leq 10^7$).

Output

Print the answer required above.

Examples

standard input	standard output
12121	12121 YES
160	61 NO

Primes from 1 to n

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

Given a number N . Print all **prime** numbers between **1** and N inclusive.

A **prime** number is a number that is greater than **1** and has only two factors which are **1** and **itself**.

In other words : **prime number divisible only by 1 and itself**.

Be careful that 1 is not prime .

The first few **prime** numbers are

2 3 5 7 11

13 17 19 23 29 31

37 41 43 47 53 59

67 71 73 79 83

Input

Only one line containing a number N ($2 \leq N \leq 10^3$).

Output

Print all prime numbers between **1** and N (inclusive) separated by a space.

Example

standard input	standard output
10	2 3 5 7

Divisors

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

Given a number N . Print all the **divisors** of N in ascending order.

Input

Only one line containing a number N ($1 \leq N \leq 10^4$).

Output

Print all **positive divisors** of N , one number per line.

Example

standard input	standard output
6	1 2 3 6
7	1 7
4	1 2 4

Note

Divisor of Number is A number that divides the integer exactly (no remainder).

In other words the division works perfectly with no fractions or remainders involved.

Examples:

- 3 is a divisor of **12**, because $12 \div 3 = 4$ exactly
- 4 is a divisor of **12**, because $12 \div 4 = 3$ exactly.
- 5 is not a divisor of **12**, because $12 \div 5 = 2$ with a remainder of 2.

a **divisor** is also a **factor** of the original integer.

GCD

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

Given two numbers A and B . Print the **greatest common divisor** between (A, B) .

Note: The greatest common divisor (**GCD**) of two or more integers, which are not all zeroes, **is the largest positive integer that divides each of the integers.**

For example:

the **GCD** of **8** and **12** is **4**.

because the numbers that divides both **8** and **12** are **(1,2,4)** and **4** is the largest one .

Input

Only one line containing two numbers A and B ($1 \leq A, B \leq 10^3$).

Output

Print the **GCD** of A and B .

Examples

standard input	standard output
12 8	4
3 7	1
3 7	1
5 10	5

Note

What is the greatest common divisor of **54 and 24**?

*The number 54 can be expressed as a product of two integers in several different ways:

$$54 * 1 = 27 * 2 = 18 * 3 = 9 * 6 \dots$$

Thus the **divisors of 54** are: **1,2,3,6,9,18,27,54**

Similarly, the divisors of 24 are: **1,2,3,4,6,8,12,24**

The numbers that these two lists share in common are **the common divisors of 54 and 24:**

1,2,3,6

The greatest of these is 6. That is, the greatest common divisor of 54 and 24. One writes:

$$\text{gcd}(54,24) = 6.$$

Lucky Numbers

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

Given two numbers A and B . Print all **lucky numbers** between A and B **inclusive**.

Note:

The **Lucky number** is any positive number that its decimal representation contains only **4** and **7**.

For example: numbers **4, 7, 47** and **744** are **lucky** and numbers **5, 17** and **174** are **not**.

Input

Only one line containing two numbers A and B ($1 \leq A \leq B \leq 10^5$).

Output

Print all **lucky numbers** between A and B **inclusive** separated by a space. If there is **no lucky number** print **-1**.

Examples

standard input	standard output
4 20	4 7
8 15	-1

Numbers Histogram

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

Given **3** lines of input described as follow:

1. First line contains a symbol S .
2. Second line contains a number N .
3. Third line contains N numbers.

For each number X_i in the N numbers print a new line that contains the symbol S repeated X_i time.

Input

The first line contains a symbol S can be $(+, -, *, /)$.

The second line an number N ($1 \leq N \leq 50$).

The third line contains N numbers ($1 \leq X_i \leq 100$).

Output

Print the answer required above.

Example

standard input	standard output
+	+++++
5	++
5 2 4 3 7	++++
	+++
	+++++++

Note

Don't print any extra spaces after symbol S .

Pyramid

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

Given a number N . Print a left angled triangle that has N rows.

For more clarification see the example below.

Input

Only one line containing a number N ($1 \leq N \leq 99$).

Output

Print the answer according to the required above.

Example

standard input	standard output
4	* ** *** ****

Note

Don't print any extra spaces after symbol `"*"`.

Shape1

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

Given a number N . Print a face down right angled triangle that has N rows.

For more clarification see the example below.

Input

Only one line containing a number N ($1 \leq N \leq 99$).

Output

Print the answer according to the required above.

Example

standard input	standard output
4	**** *** ** *

Note

Don't print any extra spaces after symbol `"*"`.

Digits

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

Given a number N . Print the **digits of that number** from right to left separated by space.

Input

First line contains a number T ($1 \leq T \leq 10$) number of test cases.

Next T lines will contain a number N ($0 \leq N \leq 10^9$)

Output

For each test case print a single line contains the **digits of the number** separated by space.

Example

standard input	standard output
4	1 2 1
121	9 3
39	6 5 4 3 2 1
123456	0 0 2 1
1200	

Sequence of Numbers and Sum

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

Given multiple lines each line contains two numbers N and M .

For each line print a single line contains:

- The numbers between N and M inclusive separated by single space.
- The message “ sum =”.
- The **summation** of all numbers between N and M inclusive.

Note: The program should be *TERMINATED* as soon as any of these two numbers is less than or equal to zero and don't print any thing.

For more clarification see the examples below.

Input

The input contains multiple line.

Each line contains two numbers N and M ($-100 \leq N, M \leq 100$).

It's **guaranteed** that the last line of the input will contain a number that is less than or equal to zero.

Output

For each line print the answer according to the required above in a single line.

Examples

standard input	standard output
5 2 5 7 5 -1	2 3 4 5 sum =14 5 6 7 sum =18
5 2 6 3 5 0	2 3 4 5 sum =14 3 4 5 6 sum =18

Note

M may be **greater** than N and **Vice Versa**.

Sum of Consecutive Odd Numbers

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

Given two numbers X and Y . Print the **sum** of all **odd** numbers between them, excluding X and Y .

Input

First line contains a number T ($1 \leq T \leq 10$) number of test cases.

Next T lines will contain two numbers X and Y ($0 \leq X, Y \leq 10^4$).

Output

Print the **sum** of all **odd** numbers between X and Y (excluding X and Y).

Example

standard input	standard output
3	0
5 6	21
10 4	12
4 9	

Shape2

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

Given a number N . Print a pyramid that has N rows.

For more clarification see the example below.

Input

Only one line containing a number N ($1 \leq N \leq 99$).

Output

Print the answer according to the required above.

Example

standard input	standard output
4	* *** ***** *****

Note

Don't print any extra spaces after symbol `"*"`.

Some Sums

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

Given three numbers N, A, B . Print the **summation** of the numbers between **1** and N whose **sum** of digits is between A and B **inclusive**.

Input

Only one line containing three numbers N, A, B ($1 \leq N \leq 10^4, 1 \leq A \leq B \leq 36$).

Output

Print a single line contains the answer according to the required above.

Examples

standard input	standard output
20 2 5	84
10 1 2	13
100 4 16	4554

Note

In the first simple:

Among the numbers not greater than **20**, the numbers whose sums of digits are between **2** and **5**, are: **2,3,4,5,11,12,13,14** and **20**.

So the answer is: **84**.

PUM

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

Given a number N . Print N lines that describes PUM game.

For more clarification see the examples.

Input

Only one line containing a number N ($1 \leq N \leq 20$).

Output

Print the answer according to the required above.

Examples

standard input	standard output
7	1 2 3 PUM 5 6 7 PUM 9 10 11 PUM 13 14 15 PUM 17 18 19 PUM 21 22 23 PUM 25 26 27 PUM
3	1 2 3 PUM 5 6 7 PUM 9 10 11 PUM

Note

Don't print any extra spaces.

Shape3

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

Given a number N . Print a diamond that has $2N$ rows.

For more clarification see the example below.

Input

Only one line containing number N ($1 \leq N \leq 99$).

Output

Print the answer according to the required above.

Example

standard input	standard output
4	* *** ***** ***** ***** ***** *** *

Note

Don't print any extra spaces after symbol "*".

Convert To Decimal 2

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 64 megabytes

Given a number N . Print the result of doing the following operation on N :

- Convert N to its **binary** representation.
- Count number of **ones** in the above **binary** representation.
- Print the equivalent **decimal** number that its **binary** representation has only the number of ones that were counted above.

For example: $(10)_{decimal} = (1010)_{binary}$ has **2** ones "11" after converting "11" to decimal number it will become **3**.

Input

First line contains a number T ($1 \leq T \leq 10$) number of test cases.

Next T lines will contain a number N ($1 \leq N \leq 2^31 - 1$).

Output

For each test case print a single line contains the answer according to the required above.

Example

standard input	standard output
3	3
10	7
7	1
8	

Note

To convert decimal number to binary :

A decimal integer can be converted to binary by dividing it by 2.

Take the quotient, and keep dividing it by 2, until you reach zero.

Each time you perform this division, take note of the remainder. Now reverse the remainders list, and you get the number in binary form

Example to convert 29 to binary

Step	Operation	Result	Remainder
Step 1	$29 / 2$	14	1
Step 2	$14 / 2$	7	0
Step 3	$7 / 2$	3	1
Step 4	$3 / 2$	1	1
Step 5	$1 / 2$	0	1

for more details visit this <https://flaviocopes.com/convert-decimal-to-binary/>

To convert from binary to Decimal :

Binary Number – 11101_2

Calculating Decimal Equivalent –

Step	Binary Number	Decimal Number
Step 1	11101_2	$((1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0))_{10}$
Step 2	11101_2	$(16 + 8 + 4 + 0 + 1)_{10}$
Step 3	11101_2	29_{10}

Second Test Case :

$(7)_{decimal} = (111)_{binary}$ has **3** ones "111" after converting "111" to decimal number it will become **7**.

Third Test Case :

$(8)_{decimal} = (1000)_{binary}$ has **1** one "1" after converting "1" to decimal number it will become **1**.

Easy Fibonacci

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

Given a number N . Print first N numbers of the **Fibonacci** sequence.

Note: In order to create the **Fibonacci** sequence use the following function:

- $\text{fib}(1) = 0$.
- $\text{fib}(2) = 1$.
- $\text{fib}(n) = \text{fib}(n - 1) + \text{fib}(n - 2)$.

Input

Only one line containing a number N ($1 \leq N \leq 45$).

Output

Print the first N numbers from the **Fibonacci** Sequence .

Example

standard input	standard output
7	0 1 1 2 3 5 8

Note

For more information visit Fibonacci: <https://www.mathsisfun.com/numbers/fibonacci-sequence.html>.

Three Numbers

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 256 megabytes

Given two numbers K and S . Determine how many **different values** of X, Y and Z such that $(0 \leq X, Y, Z \leq K)$ and $X + Y + Z = S$.

Input

Only one line containing two numbers K and S ($0 \leq K \leq 3000, 0 \leq S \leq 3K$).

Output

Print the answer required above.

Examples

standard input	standard output
2 1	3
9 4	15

Note

In the first test case all values of X, Y, Z that satisfy the conditions are :

0 0 1

0 1 0

1 0 0

In the second test case all values of X, Y, Z that satisfy the conditions are :

X Y Z	X Y Z	X Y Z	X Y Z
0 0 4	0 4 0	2 0 2	3 1 0
0 1 3	1 0 3	2 1 1	4 0 0
0 2 2	1 1 2	2 2 0	1 3 0
0 3 1	1 2 1	3 0 1	