

Sheet #7 (Recursion)

A. Print Recursion

1 second🕒, 256 megabytes

Given a number N . Print "I love Recursion" N times.

Note: Solve this problem using recursion.

Input

Only one line containing a number N ($1 \leq N \leq 100$) .

Output

Print "I love Recursion" N times.

input
3
output
I love Recursion I love Recursion I love Recursion

B. Print from 1 to N

1 second🕒, 64 megabytes

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

Note: Solve this problem using recursion.

Input

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

Output

Print N lines according to the required above.

input
5
output
1 2 3 4 5

C. Print from N to 1

1 second🕒, 64 megabytes

Given a number N . Print all numbers from N to **1** separated by a single space.

Note: Solve this problem using recursion.

Input

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

Output

Print from N to **1** separated by a single space.

input
4
output
4 3 2 1

Make sure don't print any leading or trailing spaces.

D. Print Digits using Recursion

1 second🕒, 256 megabytes

Given a number N . Print the digits of N separated by a space.

Note: Solve this problem using recursion.

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.

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

E. Base Converssion

1 second🕒, 256 megabytes

Given a number N . Print the binary equivalent of N .

Note: Solve this problem using recursion.

Input

First line contains a number T ($1 \leq T \leq 10^4$) number of test cases.

Next T lines will contain a number N ($1 \leq N \leq 10^9$).

Output

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

input
2 10 3
output
1010 11

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/converting-decimal-to-binary/>.

F. Print Even Indices

1 second🕒, 256 megabytes

Given a number N and an array A of N numbers. Print the numbers in **even** indices in a reversed order.

Note:

- Assume array A is 0-based indexing.
- Solve this problem using recursion.

Input

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

Second line contains N numbers ($- 10^9 \leq A_i \leq 10^9$).

Output

Print numbers in even indices in a reversed order separated by spaces.

input
4 1 4 2 7
output
2 1

input
7 1 5 8 2 3 9 11
output
11 3 8 1

G. Pyramid

1 second🏆, 256 megabytes

Given a number N . Print a pyramid of height N .

Note: Solve this problem using recursion.

Input

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

Output

Print the pyramid in N lines.

input
1
output
*

input
2
output
* ***

input
3
output
* *** *****

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

Don't print any extra space after '*'.

H. Inverted Pyramid

1 second🏆, 256 megabytes

Given a number N . Print an inverted pyramid of height N .

Note: Solve this problem using recursion.

Input

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

Output

Print the pyramid in N lines.

Don't print any extra space after '*'.

input
1
output
*

input
2
output
*** *

input
3

output
***** *** *

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

I. Count Vowels

1 second🏆, 256 megabytes

Given a string S . Print number of vowels in the string.

Note:

- Vowel letters: ['a', 'e', 'i', 'o', 'u'].
- Vowel letters could be capital or small.
- Solve this problem using recursion.

Input

Only one line containing a string S ($1 \leq |S| \leq 200$) where $|S|$ is the length of the string and it consists only of **capital** ,**small** letters and **spaces**.

Output

Print number of vowels in string S .

input
Data Structure Lab
output
6

J. Factorial

1 second🏆, 64 megabytes

Given a number N . Print factorial of N .

Note: Solve this problem using recursion.

Input

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

Output

Print the factorial of the number N .

input
5
output
120

K. Max Number

1 second🏆, 64 megabytes

Given a number N and an array A of N numbers. Print the **maximum** value in this array.

Note: Solve this problem using recursion.

Input

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

Second line contains N numbers ($- 10^9 \leq A_i \leq 10^9$).

Output

Print the **maximum** value in this array.

input
5 1 -3 5 4 -6
output
5

L. Summation

1 second🏆, 256 megabytes

Given a number N and an array A of N numbers. Print the **summation** of the array elements.

Note: Solve this problem using recursion.

Input

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

Second line contains N numbers ($-10^9 \leq A_i \leq 10^9$).

Output

Print the **summation** of the N numbers.

input
4 1 4 2 7
output
14

input
4 5 5 5 5
output
20

M. Suffix Sum

1 second🕒, 256 megabytes

Given two numbers N and M , and an array A of N numbers. Calculate the sum of the last M numbers.

Note: solve this problem using recursion.

Input

First line contains two numbers N and M ($1 \leq M \leq N \leq 10^5$).

Second line contains N numbers ($-10^9 \leq A_i \leq 10^9$).

Output

Print the sum of the last M numbers of the given array.

input
5 3 1 8 2 10 3
output
15

N. Sum of a Matrix

1 second🕒, 256 megabytes

Given two matrices A and B of size $R * C$. Print the **summation** of A and B .

Note: Solve this problem using recursion.

Input

First line contains two numbers R and C ($1 \leq R, C \leq 100$). number of rows and number of columns respectively.

Next R lines will contain C numbers ($-100 \leq A_{i,j} \leq 100$) matrix A numbers.

Next R lines will contain C numbers ($-100 \leq B_{i,j} \leq 100$) matrix B numbers.

Output

Print the **summation** result.

input
2 3 1 2 3 4 5 6 1 3 5 7 9 11
output
2 5 8 11 14 17

O. Fibonacci

1 second🕒, 256 megabytes

Given a number N . Print the value of the N_{th} Fibonacci number.

$$F(n) \begin{cases} 0 & \text{if } n = 1 \\ 1 & \text{if } n = 2 \\ F(n-1) + F(n-2) & \text{if } n \geq 3 \end{cases}$$

Note: Solve this problem using recursion.

Input

Only one line containing a number N ($1 \leq N \leq 30$).

Output

Print the value of the N_{th} Fibonacci number.

input
5
output
3

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

P. Log2

1 second🕒, 256 megabytes

Given a number N . Print $\lfloor \log_2(N) \rfloor$.

Note: Solve this problem using recursion.

Input

Only one line containing a number N ($1 \leq N \leq 10^{18}$).

Output

Print the answer required above.

input
1
output
0

input
8
output
3

Q. 3n + 1 sequence

1 second🕒, 256 megabytes

Given a number n , you should print the length of the $3n + 1$ sequence starting with n .

The sequence is constructed as follows:

- If the number n is odd, the next number will be $3n + 1$.
- If the number n is even, the next number will be $n/2$.

For example, the $3n + 1$ sequence of 3 is {3, 10, 5, 16, 8, 4, 2, 1} and its length is 8.

Note: Solve this problem using recursion.

Input

Only one line containing a number n ($1 \leq n \leq 10^5$).

Output

Print the length of $3n + 1$ **sequence** of the given n .

input
1
output
1

input
2
output
2

input
3
output
8

R. Palindrome Array

1 second🕒, 256 megabytes

Given a number N and an array A of N numbers. Determine if it's **palindrome** or **not**.

Note:

An array is called **palindrome** if it reads the same backward and forward, for example, arrays $\{ 1 \}$ and $\{ 1,2,3,2,1 \}$ are **palindromes**, while arrays $\{ 1,12 \}$ and $\{ 4,7,5,4 \}$ are **not**.

NOTE: Solve it using recursion.

Input

First line contains a number N ($1 \leq N \leq 10^5$) number of elements.

Second line contains N numbers ($1 \leq A_i \leq 10^9$).

Output

Print **"YES"** (without quotes) if A is a **palindrome** array, otherwise, print **"NO"** (without quotes).

input
5 1 3 2 3 1
output
YES

input
4 1 2 3 4
output
NO

S. Array Average

1 second🕒, 256 megabytes

Given a number N and an array A of N numbers. Calculate the average of these numbers.

Note: solve this problem using recursion.

Input

First line contains a number N ($1 \leq N \leq 100$) number of elements.

Second line contains N numbers ($-10^9 \leq A_i \leq 10^9$).

Output

Print the calculated **average**, with **6** digits after the decimal point.

input
5 1 2 3 4 5
output
3.000000

T. Combination

2 seconds🕒, 64 megabytes

Given two numbers N and R . Print the ${}^N C_R$ value.

Note: Solve this problem using recursion.

Input

Only one line contains two numbers N and R ($0 \leq N, R \leq 30$).

Output

Print the ${}^N C_R$ value.

input
4 2
output
6

input
3 3
output
1

For more information visit combination:

<https://www.mathsisfun.com/combinatorics/combinations-permutations.html>

U. Knapsack

2 seconds🕒, 256 megabytes

There are N items numbered from **1** to N . The i_{th} item has a weight of w_i and a value of v_i .

You have to choose some items out of the N items and carry them home in a knapsack. The capacity of the knapsack is W which donate the **maximum** weight that can be carried inside the knapsack. In other words, W means the total summation of all weights of items that can be carried in the knapsack.

Print **maximum** possible sum of values of items that you can take home.

Note: Solve this problem using recursion.

Input

First line contains two numbers N and W ($1 \leq N \leq 20, 1 \leq W \leq 100$) number of items and the capacity of the knapsack.

Next N lines will contain two numbers w_i and v_i ($1 \leq w_i \leq 50, 1 \leq v_i \leq 1000$)

Output

Print **maximum** possible sum of values of items that you can take home.

input
3 8 3 30 4 50 5 60
output
90

input
6 15 6 5 5 6 6 4 6 6 3 5 7 2
output
17

V. Creating Expression1

1 second🕒, 256 megabytes

Given two numbers N , X and an array A of N numbers. Determine if there is a way to put '+' or '-' signs between every two numbers in the array A in order to make an expression that is equal to X .

Note: Solve this problem using recursion.

Input

First line contains two numbers N and X ($1 \leq N \leq 20, -10^9 \leq X \leq 10^9$).

Second line contains N distinct numbers $A_1, A_2, \dots A_N$ ($1 \leq A_i \leq 10^5$).

Output

Print **"YES"** if you can put '+' or '-' signs between every two number to create an expression that is equal to X otherwise, print **"NO"**.

input
5 5 1 2 3 4 5
output
YES

Next N lines each of them will contain M numbers ($-10^5 \leq A_{i,j} \leq 10^5$).

Output
Print the **maximum sum** of numbers can be obtained.

input
3 3 5 2 4 1 3 5 9 2 7
output
24

Y. Number of Ways

1 second🏆, 256 megabytes

Given two numbers S and E where S denotes a start point and E denotes an end point. Determine how many possible ways to reach point E if you can move either **1** step, **2** steps or **3** steps at a time.

Note: Solve this problem using recursion.

Input
Only one line contains two numbers S and E ($1 \leq S \leq E \leq 15$).

Output
Print the answer required above.

input
2 5
output
4

In the first example:

There are **4** ways to reach from point **2** to point **5** as follows: **[2, 3, 4, 5]**, **[2, 3, 5]**, **[2, 4, 5]** and **[2, 5]**.

Z. Left Max

1 second🏆, 256 megabytes

Given a number N and an array A of N numbers, print the maximum in the range from 1 to i for each $i \leq N$.

Note: Solve this problem using recursion.

Input
First line contains a number N ($1 \leq N \leq 10^5$) number of elements.

Second line contains N numbers ($-10^9 \leq A_i \leq 10^9$).

Output
Print N numbers, the maximum from index 1 to index i .

input
5 4 3 5 7 3
output
4 4 5 7 7

input
5 2 1 2 3 4 5
output
NO

In the first example: **1 - 2 - 3 + 4 + 5 = 5**

W. Reach Value

1 second🏆, 256 megabytes

Given a number N . Initially you have a value equal **1** and you can perform one of the following operation any number of times:

1. Multiply your current value by 10.
2. Multiply your current value by 20.

Determine if your value can reach N or not.

Note: Solve this problem using recursion.

Input
First line contains a number T ($1 \leq T \leq 100$) number of test cases.

Next T lines will contain a number N ($1 \leq N \leq 10^{12}$).

Output
For each test case print **"YES"** if your value can reach exactly N otherwise, print **"NO"**.

input
5 1 2 10 25 200
output
YES NO YES NO YES

X. The maximum path-sum

1 second🏆, 256 megabytes

Given a matrix A of size $N * M$. Print the **maximum sum** of numbers that can be obtained when you take a path from $A_{1,1}$ to $A_{N,M}$.

If you stay in $A_{i,j}$ you can only go to :

- $A_{i+1,j}$ if and only if $i \leq N$
- $A_{i,j+1}$ if and only if $j \leq M$

Note: Solve this problem using recursion.

Input
First line contains two numbers N and M ($1 \leq N, M \leq 10$) N donates number of rows and M donates number of columns.