Q1. First Factorial

Have the function ***FirstFactorial(num)*** take the ***num*** parameter being passed and return the factorial of it. For example: if ***num*** = 4, then your program should return (4 * 3 * 2 * 1) = 24. For the test cases, the range will be between 1 and 18 and the input will always be an integer.

Examples

Input: 4	Output: 24
Input: 8	Output: 40320

Q2. Array Addition I

Have the function ***ArrayAdditionI(arr)*** take the array of numbers stored in ***arr*** and return the string true if any combination of numbers in the array (excluding the largest number) can be added up to equal the largest number in the array, otherwise return the string false. For example: if ***arr*** contains [4, 6, 23, 10, 1, 3] the output should return true because 4 + 6 + 10 + 3 = 23. The array will not be empty, will not contain all the same elements, and may contain negative numbers.

Examples

Input: [5,7,16,1,2]	Output: false
Input: [3,5,-1,8,12]	Output: true

Q3. sum of two numbers

sum(arr) :

Examples

Input: [2,3]	Output: 5
Input: [3,5]	Output: 8

Q4. Word Count

Have the function ***WordCount(str)*** take the ***str*** string parameter being passed and return the number of words the string contains (e.g. "Never eat shredded wheat or cake" would return 6). Words will be separated by single spaces.

Examples

Input: "Hello World"	Output: 2
Input: "one 22 three"	Output: 3

Q5. Run Length

Have the function ***RunLength(str)*** take the ***str*** parameter being passed and return a compressed version of the string using the Run-length encoding algorithm. This algorithm works by taking the occurrence of each repeating character and outputting that number along with a single character of the repeating sequence. For example: "wwwggopp" would return 3w2g1o2p. The string will not contain any numbers, punctuation, or symbols.

Examples

Input: "aabbcde"	Output: 2a2b1c1d1e
Input: "wwwbbbw"	Output: 3w3b1w