1 . Create a function which takes in a word and spells it out, by consecutively adding letters until the full word is completed. (Score 3)

Examples

spelling("bee") ➞ ["b", "be", "bee"]

spelling("happy") ➞ ["h", "ha", "hap", "happ", "happy"]

spelling("eagerly") ➞ ["e", "ea", "eag", "eage", "eager", "eagerl", "eagerly"]

2. Given an array and an integer n, return the sum of the first n numbers in the array. (Score 3)

Worked Example

sliceSum([9, 8, 7, 6], 3) ➞ 24

// The parameter n is specified as 3.

// The first 3 numbers in the list are 9, 8 and 7.

// The sum of these 3 numbers is 24 (9 + 8 + 7).

// Return the answer.

Examples

sliceSum([1, 3, 2], 2) ➞ 4

sliceSum([4, 2, 5, 7], 4) ➞ 18

sliceSum([3, 6, 2], 0) ➞ 0

3 . Given an input string, reverse the string word by word, the first word will be the last, and so on. (score 2)

Examples

reverseWords(" the sky is blue") ➞ "blue is sky the"

reverseWords("hello   world! ") ➞ "world! hello"

reverseWords("a good example") ➞ "example good a"

Notes

- A word is defined as a sequence of non-space characters.
- The input string may contain leading or trailing spaces. However, your reversed string should not contain leading or trailing spaces.
- You need to reduce multiple spaces between two words to a single space in the reversed string.
- Try to solve this in linear time.

4 . Given a string, return true if its length is even or false if the length is odd. (Score 2)

Examples

oddOrEven("apples") ➞ true

// The word "apples" has 6 characters.

// 6 is an even number, so the program outputs true.

oddOrEven("pears") ➞ false

// "pears" has 5 letters, and 5 is odd.

// Therefore the program outputs false.

oddOrEven("cherry") ➞ true