

#1

Super Reduced String

Reduce a string of lowercase characters in range `ascii['a'..'z']` by doing a series of operations. In each operation, select a pair of adjacent letters that match, and delete them.

Delete as many characters as possible using this method and return the resulting string. If the final string is empty, return Empty String

Given a string in delete all adjacent letters and return the result

Examples:

"aab" -> b (remove aa as the are similar)

"abba" -> " Empty String"

"aaabccddd" -> "abd"

#2

Convert Hours and Minutes into Seconds

Write a function that takes two integers (hours, minutes), converts them to seconds, and adds them.

Examples

`convert(1, 3) → 3780`

`convert(2, 0) → 7200`

`convert(0, 0) → 0`

#3

Repeating Letters

Create a function that takes a string and returns a string in which each character is repeated once.

Examples

`doubleChar("String")` → `"SSttrriinnngg"`

`doubleChar("Hello World!")` → `"HHeellllloo WWoorrlldd!!"`

`doubleChar("1234!_ ")` → `"11223344!!__ "`

#4

Equality of 3 Values

Create a function that takes three integer arguments (a, b, c) and returns the amount of integers which are of equal value.

Examples

`equal(3, 4, 3)` → 2

`equal(1, 1, 1)` → 3

`equal(3, 4, 1)` → 0

#5

factorial

Using only recursion create a function that returns the factorial of input number

Examples

`factorial(5)` → 120

`factorial(3)` → 6

`factorial(2)` → 2

#6

Count Ones in a 2D Array

Create a function to count the number of (1)s in a 2D array.

Examples

`count_ones([[1, 0],[0, 0]])` → 1

`count_ones([[1, 1, 1],[0, 0, 1],[1, 1, 1]])` → 7

`count_ones([[1, 2, 3],[0, 2, 1],[5, 7, 33]])` → 2