Question - 1
Multiples of 3 and 5

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.
Write a function `getSum()` to return the sum of all the multiples of 3 or 5 below a given number passed as parameter to your function.

Note: The function stub is already available to you - you only have to add the implementation within the stub.

Question - 2
Sum of two numbers

You are given two lists representing one non-negative number each. The digits of the number represented by the list is stored in reverse order in the list.
Write a function to add the numbers and return it as a list. The arguments to your function will be the aforementioned two lists.

For example,

Input: `[2, 4, 6],[8, 0, 9]`
Output: `[0, 5, 5, 1]`
Explanation: `642 + 908 = 1550`

Question - 3
Matching brackets

Given a string containing only the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, find out if the input string is valid.

The input string is valid if:

- Open brackets are closed by the same type of brackets.
- Open brackets are closed in the correct order.

For example,

Input: `"{}"`
Output: `true`

Input: `"{[]}"`
Output: `true`

Input: `"{}[]"`
Output: `true`

Input: `"{[}]"`
Output: `false`

Implement a function `isValid()` that accepts the aforementioned string and returns the total number of brackets when the string is valid and -1 when it is not.

Question - 4
Simplify path

Given an absolute file path (unix / linux style), reduce it.

For example,

Input: `"/root/"`
Output: `"/root"`

Input: `"/x/./y/../z/"`
Output: `"/x/z"`

`.` represents the current directory and `..` represents parent directory.
Consider `"/../"` as `"/"` and `"/x//y/"` as `"/x/y"`.

Write a function `reducePath()` that accepts a path string and return the reduced path.

Question - 5
Valid Palindrome

Given a string, figure out if it is a palindrome considering only alphanumeric characters and ignoring cases.

For example,

Input: `"A man, a plan, a canal: Panama"`
Output: `true`

Input: `"race a car"`
Output: `false`

Write a function `isPalindrome()` that returns the total length of the original input string when it is a palindrome according to the above mentioned rules or -1 when it is not.