

1 Two Sum

Easy

Given an array of integers, return **indices** of the two numbers such that they add up to a specific target.

You may assume that each input would have **exactly** one solution, and you may not use the *same* element twice.

Example:

```
Given nums = [2, 7, 11, 15], target = 9,
```

```
Because nums[0] + nums[1] = 2 + 7 = 9,  
return [0, 1].
```

2 Add Two Numbers

Medium

You are given two **non-empty** linked lists representing two non-negative integers. The digits are stored in **reverse order** and each of their nodes contain a single digit. Add the two numbers and return it as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

Example

```
Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)
```

```
Output: 7 -> 0 -> 8
```

```
Explanation: 342 + 465 = 807.
```

3 Longest Substring Without Repeating Characters

Medium

Given a string, find the length of the **longest substring** without repeating characters.

Examples:

Given "abcabcbb", the answer is "abc", which the length is 3.

Given "bbbbbb", the answer is "b", with the length of 1.

Given "pwwkew", the answer is "wke", with the length of 3. Note that the answer must be a **substring**, "pwke" is a *subsequence* and not a substring.

4 Median of Two Sorted Arrays

Hard

There are two sorted arrays **nums1** and **nums2** of size m and n respectively.

Find the median of the two sorted arrays. The overall run time complexity should be $O(\log(m+n))$.

Example 1:

```
nums1 = [1, 3]
nums2 = [2]

The median is 2.0
```

Example 2:

```
nums1 = [1, 2]
nums2 = [3, 4]

The median is (2 + 3)/2 = 2.5
```

5 Longest Palindromic Substring

Medium

Given a string **s**, find the longest palindromic substring in **s**. You may assume that the maximum length of **s** is 1000.

Example 1:

```
Input: "babad"
Output: "bab"
Note: "aba" is also a valid answer.
```

Example 2:

```
Input: "cbbd"
Output: "bb"
```

6 ZigZag Conversion

Medium

The string "PAYPALISHIRING" is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)

```
P   A   H   N
A P L S I I G
Y   I   R
```

And then read line by line: "PAHNAPLSIIGYIR"

Write the code that will take a string and make this conversion given a number of rows:

```
string convert(string s, int numRows);
```

Example 1:

```
Input: s = "PAYPALISHIRING", numRows = 3
Output: "PAHNAPLSIIGYIR"
```

Example 2:

```
Input: s = "PAYPALISHIRING", numRows = 4
Output: "PINALSIGYAHRPI"
Explanation:
```

```
P       I       N
A    L S   I G
Y A   H R
P       I
```

7 Reverse Integer

Easy

Given a 32-bit signed integer, reverse digits of an integer.

Example 1:

```
Input: 123
Output: 321
```

Example 2:

```
Input: -123
Output: -321
```

Example 3:

```
Input: 120
Output: 21
```

Note:

Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range: $[-2^{31}, 2^{31} - 1]$. For the purpose of this problem, assume that your function returns 0 when the reversed integer overflows.

8 String to Integer (atoi)

Medium

Implement `atoi` which converts a string to an integer.

The function first discards as many whitespace characters as necessary until the first non-whitespace character is found. Then, starting from this character, takes an optional initial plus or minus sign followed by as many numerical digits as possible, and interprets them as a numerical value.

The string can contain additional characters after those that form the integral number, which are ignored and have no effect on the behavior of this function.

If the first sequence of non-whitespace characters in `str` is not a valid integral number, or if no such sequence exists because either `str` is empty or it contains only whitespace characters, no conversion is performed.

If no valid conversion could be performed, a zero value is returned.

Note:

- Only the space character ' ' is considered as whitespace character.
- Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range: $[-2^{31}, 2^{31} - 1]$. If the numerical value is out of the range of representable values, `INT_MAX` ($2^{31} - 1$) or `INT_MIN` (-2^{31}) is returned.

Example 1:

Input: "42"
Output: 42

Example 2:

Input: " -42"
Output: -42
Explanation: The first non-whitespace character is '-', which is the minus sign.
Then take as many numerical digits as possible, which gets 42.

Example 3:

Input: "4193 with words"
Output: 4193
Explanation: Conversion stops at digit '3' as the next character is not a numerical digit.

Example 4:

Input: "words and 987"
Output: 0
Explanation: The first non-whitespace character is 'w', which is not a numerical digit or a +/- sign. Therefore no valid conversion could be performed.

Example 5:

Input: "-91283472332"
Output: -2147483648
Explanation: The number "-91283472332" is out of the range of a 32-bit signed integer.
Therefore `INT_MIN` (-2^{31}) is returned.

9 Palindrome Number

Easy

Determine whether an integer is a palindrome. An integer is a palindrome when it reads the same backward as forward.

Example 1:

```
Input: 121
Output: true
```

Example 2:

```
Input: -121
Output: false
Explanation: From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a
```

Example 3:

```
Input: 10
Output: false
Explanation: Reads 01 from right to left. Therefore it is not a palindrome.
```

Follow up:

Could you solve it without converting the integer to a string?

10 Regular Expression Matching

Hard

Given an input string (`s`) and a pattern (`p`), implement regular expression matching with support for `.` and `*`.

`.` Matches any single character.
`*` Matches zero or more of the preceding element.

The matching should cover the **entire** input string (not partial).

Note:

- `s` could be empty and contains only lowercase letters `a-z`.
- `p` could be empty and contains only lowercase letters `a-z`, and characters like `.` or `*`.

Example 1:

Input:
`s = "aa"`
`p = "a"`
Output: false
Explanation: "a" does not match the entire string "aa".

Example 2:

Input:
`s = "aa"`
`p = "a*"`
Output: true
Explanation: `*` means zero or more of the preceding element, 'a'. Therefore, by repeating 'a' once, it becomes "aa".

Example 3:

Input:
`s = "ab"`
`p = ".*"`
Output: true
Explanation: `.*` means "zero or more (*) of any character (.)".

Example 4:

Input:
`s = "aab"`
`p = "c*a*b"`
Output: true
Explanation: c can be repeated 0 times, a can be repeated 1 time. Therefore it matches "aab".

Example 5:

Input:
`s = "mississippi"`
`p = "mis*is*p*."`
Output: false

11 Container With Most Water

Medium

Given n non-negative integers a_1, a_2, \dots, a_n , where each represents a point at coordinate (i, a_i) . n vertical lines are drawn such that the two endpoints of line i is at (i, a_i) and $(i, 0)$. Find two lines, which together with x-axis forms a container, such that the container contains the most water.

Note: You may not slant the container and n is at least 2.

12 Integer to Roman

Medium

Roman numerals are represented by seven different symbols: **I**, **V**, **X**, **L**, **C**, **D** and **M**.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, two is written as **II** in Roman numeral, just two one's added together. Twelve is written as, **XII**, which is simply **X** + **II**. The number twenty seven is written as **XXVII**, which is **XX** + **V** + **II**.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not **IIII**. Instead, the number four is written as **IV**. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as **IX**. There are six instances where subtraction is used:

- I** can be placed before **V** (5) and **X** (10) to make 4 and 9.
- X** can be placed before **L** (50) and **C** (100) to make 40 and 90.
- C** can be placed before **D** (500) and **M** (1000) to make 400 and 900.

Given an integer, convert it to a roman numeral. Input is guaranteed to be within the range from 1 to 3999.

Example 1:

Input: 3
Output: "III"

Example 2:

Input: 4
Output: "IV"

Example 3:

Input: 9
Output: "IX"

Example 4:

Input: 58
Output: "LVIII"
Explanation: C = 100, L = 50, XXX = 30 and III = 3.

Example 5:

Input: 1994
Output: "MCMXCIV"
Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

13 Roman to Integer

Easy

Roman numerals are represented by seven different symbols: **I**, **V**, **X**, **L**, **C**, **D** and **M**.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For example, two is written as **II** in Roman numeral, just two one's added together. Twelve is written as, **XII**, which is simply **X** + **II**. The number twenty seven is written as **XXVII**, which is **XX** + **V** + **II**.

Roman numerals are usually written largest to smallest from left to right. However, the numeral for four is not **IIII**. Instead, the number four is written as **IV**. Because the one is before the five we subtract it making four. The same principle applies to the number nine, which is written as **IX**. There are six instances where subtraction is used:

- I** can be placed before **V** (5) and **X** (10) to make 4 and 9.
- X** can be placed before **L** (50) and **C** (100) to make 40 and 90.
- C** can be placed before **D** (500) and **M** (1000) to make 400 and 900.

Given a roman numeral, convert it to an integer. Input is guaranteed to be within the range from 1 to 3999.

Example 1:

Input: "III"
Output: 3

Example 2:

Input: "IV"
Output: 4

Example 3:

Input: "IX"
Output: 9

Example 4:

Input: "LVIII"
Output: 58
Explanation: C = 100, L = 50, XXX = 30 and III = 3.

Example 5:

Input: "MCMXCIV"
Output: 1994
Explanation: M = 1000, CM = 900, XC = 90 and IV = 4.

14 Longest Common Prefix

Easy

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string `""`.

Example 1:

```
Input: ["flower","flow","flight"]
Output: "fl"
```

Example 2:

```
Input: ["dog","racecar","car"]
Output: ""
Explanation: There is no common prefix among the input strings.
```

Note:

All given inputs are in lowercase letters `a-z`.

15 3Sum

Medium

Given an array `nums` of n integers, are there elements a, b, c in `nums` such that $a + b + c = 0$? Find all unique triplets in the array which gives the sum of zero.

Note:

The solution set must not contain duplicate triplets.

Example:

```
Given array nums = [-1, 0, 1, 2, -1, -4],

A solution set is:
[
  [-1, 0, 1],
  [-1, -1, 2]
]
```

16 3Sum Closest

Medium

Given an array `nums` of n integers and an integer `target`, find three integers in `nums` such that the sum is closest to `target`. Return the sum of the three integers. You may assume that each input would have exactly one solution.

Example:

```
Given array nums = [-1, 2, 1, -4], and target = 1.

The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).
```

17 Letter Combinations of a Phone Number

Medium

Given a string containing digits from **2-9** inclusive, return all possible letter combinations that the number could represent.

A mapping of digit to letters (just like on the telephone buttons) is given below. Note that 1 does not map to any letters.



Example:

Input: "23"

Output: ["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"].

Note:

Although the above answer is in lexicographical order, your answer could be in any order you want.

18 4Sum

Medium

Given an array **nums** of n integers and an integer **target**, are there elements a, b, c , and d in **nums** such that $a + b + c + d = \text{target}$? Find all unique quadruplets in the array which gives the sum of **target**.

Note:

The solution set must not contain duplicate quadruplets.

Example:

Given array `nums = [1, 0, -1, 0, -2, 2]`, and `target = 0`.

A solution set is:

```
[
  [-1, 0, 0, 1],
  [-2, -1, 1, 2],
  [-2, 0, 0, 2]
]
```

19 Remove Nth Node From End of List

Medium

Given a linked list, remove the n -th node from the end of list and return its head.

Example:

Given linked list: **1->2->3->4->5**, and $n = 2$.

After removing the second node from the end, the linked list becomes **1->2->3->5**.

Note:

Given n will always be valid.

Follow up:

Could you do this in one pass?

20 Valid Parentheses

Easy

Given a string containing just the characters **'('**, **')'**, **'{'**, **'}'**, **'['** and **']'**, determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Note that an empty string is also considered valid.

Example 1:

Input: `"()"`
Output: `true`

Example 2:

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

Example 3:

Input: `"(]"`
Output: `false`

Example 4:

Input: `"([)]"`
Output: `false`

Example 5:

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