

Suggested Pre-Read

<https://kkspeed.github.io/leetcode-scala/basic-recursion.html>

Prep Questions Starts Below :

MCQ

Freshers (for interview prep)

1. Can you explain what Scala is and why it's used?
2. What are the main differences between val and var in Scala?
3. Explain what an immutable variable is. How is it useful?
4. What is a function in Scala, and how do you define one?
5. What is the difference between a function and a method in Scala?
6. How do you create a class in Scala?
7. What is an object in Scala, and how does it differ from a class?
8. Explain what a case class is and when you might use it.
9. What is a trait in Scala, and how is it different from a class?
10. How do you implement inheritance in Scala?
11. What is the purpose of the 'main' method in Scala?
12. Can you describe what pattern matching is in Scala?
13. What is a Scala list, and how do you add elements to it?
14. How do you iterate through a list in Scala?

15. What are the advantages of using immutable collections in Scala?
 16. What is an Option in Scala, and why is it useful?
 17. Explain the purpose of the 'for' comprehension in Scala.
 18. What is a tuple in Scala, and how do you access its elements?
 19. Can you describe what a higher-order function is in Scala?
 20. How do you handle exceptions in Scala?
 21. What is the REPL in Scala, and how is it used?
 22. How do you compile and run a Scala program?
 23. Explain the concept of currying in Scala.
 24. What is the difference between 'apply' and 'unapply' methods in Scala?
 25. How do you define default parameter values in a Scala function?
 26. Can you explain the concept of type inference in Scala?
 27. What are some of the benefits of using Scala over Java?
-

Intermediate (for interview prep)

1. What is the difference between `val` and `var` in Scala? Can you give simple examples?
2. Explain what an `Option` in Scala is, and why we use it.
3. What is a Scala Trait, and how is it different from a Class?
4. How do you define a function in Scala? Show a simple example.
5. What does `immutable` mean in the context of Scala collections?
6. Explain what pattern matching is in Scala and show a basic example.
7. What is the purpose of the `Unit` type in Scala?

8. How do you create a simple `for` loop in Scala?
 9. What is a Scala `case class`, and what are some of its benefits?
 10. Describe what you understand about Scala's type inference.
 11. What is the difference between `List` and `Array` in Scala?
 12. How can you define default parameter values for a function in Scala?
 13. Explain what you know about string interpolation in Scala.
 14. What does it mean for a function to be a 'first-class citizen' in Scala?
 15. Describe a simple use case for using `if/else` statements in Scala.
 16. How would you create a simple class with a constructor in Scala?
 17. What is the role of the `main` method in a Scala program?
 18. Can you explain what a higher-order function is in Scala?
 19. What are some advantages of using Scala over Java, in your opinion?
 20. Explain what you know about the concept of immutability and how Scala supports it.
 21. What are the main differences between mutable and immutable collections? When would you choose one over the other?
 22. Can you describe how to handle exceptions in Scala using `try-catch` blocks?
 23. How do you define a method within a class in Scala?
 24. What are the basic building blocks of a simple Scala application?
 25. How can you add extra behavior (methods) to a class after it's been defined in Scala, and what is this concept called?
-

Scala

Q1. Scala bytecode can run on top of Java VM. What is the fundamental difference between Java object.clone() and Scala object.copy()?

- [] One is a Java object, the other is a Scala object.
- [] clone() will copy class structures but not the data, while copy() will also copy data into new objects.
- [] There is no difference.
- [] copy() allows you to change values during the copying process; clone() does not.

Q2. What value does this code return?

```
```scala
val m1 = Map("a"->1,"b"->2,"c"->3)
m1("a")
````
```

- [] a
- [] 2
- [] b
- [] 1

Q3. What is one way to avoid low-level parallelization details?

- [] monads
- [] literal functions
- [] partially applied functions
- [] parallel collections

Q4. What do you use in ScalaTest to see a detailed diagram of error messages when a test fails?

- [] ArgumentExceptions
- [] AssertionException
- [] DiagrammedAssertions
- [] JUnit

Q5. What data type would you use to store an immutable collection of objects that contain a fixed number of varying types?

- [] Array
- [] ImmutableCollection
- [] List
- [] Tuple

Q6. After defining a function in the interpreter, Scala returns the following. What does the `()` indicate?

``

myfnc: ()Unit

``

- [] The function has no side effects.
- [] The function takes no parameters.
- [] The function returns no value.
- [] Returning unit types to the function is a closures.

Q7. What type of number is 1234.e5?

- [] hexadecimal
- [] short
- [] floating point
- [] long

Q8. When you convert a map to a list using the `toList` method of the map, the result will be of which type?

- [] `List[(String, String)]`
- [] `List[(Array, Array)]`
- [] `List[(Collection, Collection)]`
- [] `List`

Q9. What type of object does this code create?

...

`val x = (1234, "Active")`

...

- [] List

- [] Map

- [] Tuple

- [] Array

Q10. Which is a subclass of all classes?

- [] AnyVal

- [] AnyRef

- [] Method

- [] Null

[Null in Scala Standard library](<https://www.scala-lang.org/api/current/scala/Null.html>). The question is a bit incorrect - Null is a subtype of every type _except those of value classes_

Q11. For the for-yield construct, is the scope separate between for-body and yield-body?

- [] Yes and no. It is different depending on the for construct and what it does.
- [] Yes, because the for section does not expose its scope.
- [] No, because for-yield shares the same scope, even though they are within separate curly braces.
- [] Yes, because they are within different curly braces.

Example: yield-body has access to the `e` variable from the for-body

```
```scala
val a = Array(1, 2, 3, 4, 5)

for {
 e <- a if e > 2
} yield e
```
```

```

#### Q12. What is one way to implement pattern matching on methods?

- [ ] using regex
- [ ] using monads
- [ ] using string matching
- [ ] using case classes

Note: ambiguous question, it's not clear what kind of [pattern matching](<https://docs.scala-lang.org/tour/pattern-matching.html>) is meant here.

#### Q13. What is the value of z after executing this code?

...

val y = List('a','b')

val z = y::List('c')

...

- [ ] List(a,b,c)

- [ ] List(List(a, b), c)

- [ ] List(c,a,b)

- [ ] List(c,List(a,b))

#### Q14. What term is used to specify a precondition?

- [ ] assert

- [ ] require

- [ ] precondition

- [ ] mustHave

#### Q15. Which Scala type may throw an exception or a successfully computed value, and is commonly used to trap and propagate errors?

- [ ] `scala.util.ExceptionHandling`

- [ ] `scala.Catch.Throw`

- [ ] `scala.exception.TryFinally`

- [ ] `scala.util.Try`

[[scala.util.Try](https://www.scala-lang.org/api/current/scala/util/Try.html)](https://www.scala-lang.org/api/current/scala/util/Try.html)

#### Q16. What is the data type of y after this code is executed?

...

`val y = (math.floor(3.1415 * 2))`

...

- [ ] short

- [ ] double

- [ ] int

- [ ] bigInt

#### Q17. When using pattern matching, which character matches on any object?

- [ ] `%`

- [ ] `\_`

- [ ] `^`

- [ ] `~`

[[Pattern Matching](https://docs.scala-lang.org/tour/pattern-matching.html)](https://docs.scala-lang.org/tour/pattern-matching.html)

#### Q18. You have created an array using val. Can you change the value of any element of the array—and why or why not?

- [ ] Yes, the reference to the array is immutable, so the location that the array points to is immutable. The values in the array are mutable.

- [ ] The 0th element is immutable and cannot be modified. All other elements can be modified.

- [ ] Yes, val does not make arrays immutable.

- [ ] No, val makes the array and values of the array immutable.

**\*\*Explanation\*\*:**

```
```scala
```

```
val a1 = Array(1, 2, 3)
```

```
a1{1} = 3 // OK
```

```
a1 = Array(1, 3, 3) // error: reassignment to val
```

```
...
```

Q19. What is the output of this function?

```
```scala
```

```
def main () {
```

```
 var a = 0
```

```
 for (a<-1 until 5){println(a)}
```

```
...
```

- [ ] 1,2,3,4,5

- [ ] 0,1,2,3,4

- [ ] 1,2,3,4

- [ ] 2,3,4,5

#### Q20. What do you call objects with immutable state?

- [ ] singletons
- [ ] stationary objects
- [ ] functional objects
- [ ] fixed objects

\*\*Note:\*\* singletons may have mutable state

#### Q21. You have written a Scala script. How would you access command-line arguments in the script?

- [ ] use array named args
- [ ] use tuple named args
- [ ] use numbered variables with a \_ prefix for example \_1, \_2, \_3
- [ ] use numbered variables with a \$ prefix - for example \$1, \$2, \$3

#### Q22. What does this code return? `val x = 3; if (x > 2) x = 4 else x = x\*2`

- [ ] 4
- [ ] an error
- [ ] 6
- [ ] 3

#### Q23. Which statement returns a success or a failure indicator when you execute this code?

`val MyFuture = Future {runBackgroundFunction() }`

- [ ] myFuture.onComplete
- [ ] myFuture(status)
- [ ] myFuture.Finished
- [ ] complete(myFuture)

#### Q24. To denote a parameter that may be repeated, what should you place after type?

- [ ] `%
- [ ] `&
- [ ] `\_`
- [ ] `^-`

#### Q25. What is called when a superclass has more than one subclass in Scala?

- [ ] polyinheritance
- [ ] multilevel inheritance
- [ ] multimode inheritance
- [ ] hierarchical inheritance

#### Q26. One way to improve code reliability is to use `\_\_` , which will evaluate a condition and return an error if the condition is violated.

- [ ] packages
- [ ] polymorphisms
- [ ] assertions
- [ ] traits

#### Q27. Which statement about if-else-if-else statements is true?

- [ ] If the first else-if does not succeed, then no other else-ifs are tested.
- [ ] If an else-if does not succeed, then none of the remaining else-if statements or elses will be tested.
- [ ] All else-if statements are tested in all cases.
- [ ] If an else-if succeeds, then none of the remaining else-if statements or elses will be tested.

#### Q28. What do you call the process of changing the definition of an inherited method?

- [ ] recursive methods
- [ ] currying methods
- [ ] redefining methods
- [ ] overriding methods

#### Q29. To denote a parameter that may be repeated, what should you place after the type?

- [ ] `\_`
- [ ] `\*`
- [ ] `%`
- [ ] `&`

[Repeated Parameters in Scala](<https://www.baeldung.com/scala/repeated-parameters>)

#### Q30. What is the code below equivalent to?

```
```scala  
myClass.foreach(println _)  
```
```

- [ ] `myClass.foreach(println ())`
- [ ] `myClass.foreach(print NIL)`
- [ ] `myClass.loop(println ())`
- [ ] `myClass.foreach(x => println(x))`

#### Q31. What is an advantage of an immutable object?

- [ ] Immutable objects use less memory than their mutable counterparts.
- [ ] Immutable objects do not require error handling.
- [ ] Immutable objects can be used in classes, mutable objects cannot.

- [ ] Immutable objects are threadsafe.

#### Q32. You want to create an iteration loop that tests the condition at the end of the loop body. Which iteration would you use?

- [ ] do-while loop

- [ ] while loop

- [ ] for loop

- [ ] do-until loop

#### Q33. What can you use to make querying a database more efficient, by avoiding the need to parse the SQL string every time a query is executed from Scala?

- [ ] database driver

- [ ] connection

- [ ] prepared statement

- [ ] SQL view

[PreparedStatement from Java](<https://docs.oracle.com/javase/8/docs/api/java/sql/PreparedStatement.html>) which is also used in Scala

#### Q34. Which is \_not\_ a member of the collections hierarchy?

- [ ] Set

- [ ] Seq

- [ ] Hash

- [ ] Map

#### Q35. Which term makes the contents of packages available without prefixing?

- [ ] use
- [ ] include
- [ ] import
- [ ] assertion

#### Q36. If you wanted to find the remainder after division, what operator would you use?

- [ ] %
- [ ] DIV
- [ ] //
- [ ] /

#### Q37. What are defined inside a class definition?

- [ ] method
- [ ] fields and methods
- [ ] fields, methods, and packages
- [ ] fields

#### Q38. What defines methods and fields that can then be reused by mixing into classes?

- [ ] singleton
- [ ] assertion
- [ ] trait
- [ ] monad

#### Q39. When do you need to explicitly state the return type in a function definition?

- [ ] when the function has no side effects
- [ ] when the function returns a Unit type
- [ ] when the function is recursive
- [ ] when the function has side effects

#### Q40. Why would you make a field private?

- [ ] so only methods in the same file can access the field
- [ ] so only methods in the same package can access the field
- [ ] so only methods in the same class could access the field
- [ ] so only methods defined in a Java class can access the field

#### Q41. What's the difference between `.`equals` and `==`?

- [ ] They do the exact same thing
- [ ] `==` won't work on objects

- [ ] `==` cannot be applied to `String`
- [ ] `==` is a wrapper of ` `.equals()` and checks for nulls

[Source:](<https://www.programmersought.com/article/3717957705/>)

#### Q42. What is denotes the intersection between two sets?

- [ ] ||
- [ ] &&
- [ ] &
- [ ] %

[Source:](<https://docs.scala-lang.org/overviews/collections/sets.html>)

#### Q43. What do you call a function defined in a block?

- [ ] private function
- [ ] block function
- [ ] local function
- [ ] method

A function defined within a block of code, such as within a method or another function, is called a local function. This is because it is only visible and accessible within the scope of the block in which it is defined, and is not accessible outside of that block.

#### Q44. What do you call a Scala method that is parametrized by type as well as by value?

- [ ] multimode method
- [ ] polymorphic method
- [ ] closure
- [ ] collection method

#### Q45. What type of exception is thrown when a precondition is violated?

- [ ] IllegalArgumentException
- [ ] NumberFormatException
- [ ] NullPointerException
- [ ] MalformedParameterException

#### Q46. In scala what is precondition?

- [ ] a constraint on where a method may be called from
- [ ] a constraint on values passed to a methode constructor
- [ ] a class of predifined error messages
- [ ] a class of Boolean operators

#### Q47. What would you change in this code to make it execute in parallel?

...

```
val myNums = (1 to 500).toList
list.map(_ + 1)
```

...

- [ ] Change \*\*list.map\*\* to \*\*list.par.map.\*\*
- [ ] Change \*\*toList\*\* to \*\*toListPar\*\*
- [ ] Change \*\*val\*\* to \*\*val.par\*\*
- [ ] Change \*\*toList\*\* to \*\*toParallelList\*\*

#### Q48. What is a free variable?

- [ ] a variable defined outside a function
- [ ] a variable referenced in a function that is not assigned a value by that function
- [ ] a variable that has a global scope
- [ ] a variable defined in a class and available to all methods in that class

#### Q49. What's the best way to execute code in the background in a separate thread?

- [ ] AltThread
- [ ] AltFuture
- [ ] AltProcess
- [ ] Future

#### Q50. What value does this code return?

```
```scala
```

```
x= List(1,2,4); x(1)?
```

```
```
```

- [ ] (1,2,4)
- [ ] 1
- [ ] Nil
- [ ] 2

#### Q51. Which data type does Scala use instead of null for optional values?

- [ ] Nil
- [ ] Option
- [ ] Singleton
- [ ] Collection

In Scala, the Option data type is used instead of null for optional values. It is a container that can either hold a value or be empty, and it is used to represent the presence or absence of a value. This makes it safer to work with than using null, as it eliminates the risk of null pointer exceptions.

#### Q52. What is equivalent to this code?

```
```val a = "baz"
```

```
s"Foo $a?"
```

```
```
```

- [ ] "Foo " + a + "?"

#### Q53. Which expression is one way to iterate over a collection and generate a collection of each iteration's result?

- [ ] for-yield
- [ ] for-collect
- [ ] for-collect until
- [ ] collectuntil

- for-yield is one way to iterate over a collection and generate a collection of each iteration's result. The for loop with the yield keyword is used to iterate over a collection and generate a new collection with the results of each iteration.

#### Q54. Which statement accesses the third element of an array named foo?

- [ ] foo[2]
- [ ] foo(3)
- [ ] foo[3]
- [ ] foo(2)

- In many programming languages, arrays are indexed starting at 0, so the first element of the array is at index 0, the second element is at index 1, and so on. Therefore, to access the third element of an array named "foo", you would use the index 2 (since the array is indexed starting at 0). This can be done using the syntax foo[2] or foo(2) depending on the programming language. In some languages like Java, you can use foo[2] or foo[3] to access the third element.

#### Q55. What data type would you use to store an immutable collection of objects when you don't know how many members will be in the collection?

- [ ] Tuple
- [ ] List
- [ ] Object
- [ ] Array

- You would use a List data type to store an immutable collection of objects when you don't know how many members will be in the collection. Lists are indexed collections of elements that can be accessed by their position in the list, and they are commonly used to store collections of items that need to be processed in order. Additionally, Lists are immutable, which means that their elements cannot be modified once they have been created, making them ideal for use cases where data integrity is important.

#### Q56. From where do all classes in Scala inherit?

- [ ] AnyRef

- [ ] AnyColl

- [ ] AnyVal

- [ ] AnyClass

- All classes in Scala inherit from the AnyRef class by default. AnyRef is the base class for all reference types in Scala, and it is equivalent to the java.lang.Object class in Java. AnyVal is the base class for all value types in Scala, and Any is the base class for all types in Scala.

#### Q57. In Scala, what is a precondition?

- [ ] A class of boolean operators

- [ ] A class of predefined error messages

- [ ] A constraint on values passed to a method or constructor

- [ ] A constraint on where a method may be called from

- A precondition in Scala is a constraint on the input values passed to a method or constructor, specifying the conditions that must be met for the method or constructor to execute correctly. It is used to check the validity of input values before the method

or constructor is executed, and can be used to ensure that the method or constructor is called with the correct arguments.

#### Q58. Which code sample will print the integers 1 through 4, each on a separate line?

- [ ] `for(i <- 0 to 4) println(i)`
- [ ] `for(i <- 0 to 3) println(i+1)`
- [ ] `for(i <- 1 to 8 if i < 5) println(i)`
- [ ] `for(i <- 1 to 4) println(i)`

- The correct answer is `for(i <- 0 to 3) println(i+1)` because it will start the iteration from 0 and end at 3, incrementing the value by 1 each time and printing it.

#### Q59. Which operator should you use to take the intersection of two sets?

- [ ] `&`
- [ ] `||`
- [ ] `&&`
- [ ] `%`

- The `&` or `intersect` method can be used to take the intersection of two sets in Scala.

#### Q60. Which data type does Scala use instead of null for optional values?

- [ ] `Nil`
- [ ] `Option`
- [ ] `Singleton`
- [ ] `Collection`

- In Scala, the Option data type is used instead of null for optional values. It is a container that can either hold a value or be empty, and it is used to represent the presence or absence of a value. This makes it safer to work with than using null, as it eliminates the risk of null pointer exceptions.

#### Q61. What is the difference between a Scala trait and an interface?

- [ ] A trait can have concrete implementations, while an interface cannot.
  - [ ] An interface can have concrete implementations, while a trait cannot.
  - [ ] Traits can be mixed together, while interfaces cannot.
  - [ ] Interfaces can be mixed together, while traits cannot.
- 

## 1. Two Sum

Easy

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to target*.

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

You can return the answer in any order.

Example 1:

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

Output: `[0,1]`

Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

Example 2:

Input: nums = [3,2,4], target = 6

Output: [1,2]

Example 3:

Input: nums = [3,3], target = 6

Output: [0,1]

Constraints:

- $2 \leq \text{nums.length} \leq 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$
- $-10^9 \leq \text{target} \leq 10^9$
- Only one valid answer exists.

Follow-up: Can you come up with an algorithm that is less than  $O(n^2)$  time complexity?

---

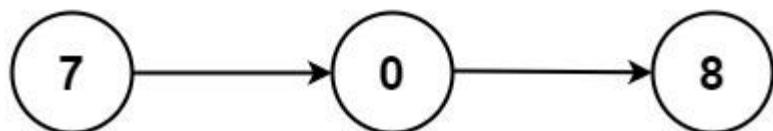
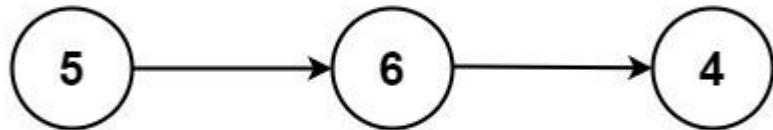
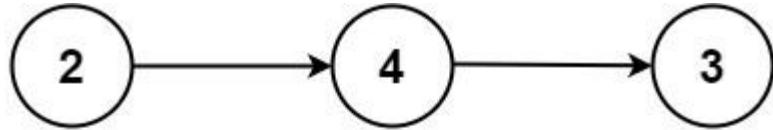
## 2. Add Two Numbers

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 contains a single digit. Add the two numbers and return the sum as a linked list.

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

Example 1:



Input:  $I1 = [2,4,3]$ ,  $I2 = [5,6,4]$

Output: [7,0,8]

Explanation:  $342 + 465 = 807$ .

Example 2:

Input:  $I1 = [0]$ ,  $I2 = [0]$

Output: [0]

Example 3:

Input:  $I1 = [9,9,9,9,9,9,9]$ ,  $I2 = [9,9,9,9]$

Output: [8,9,9,9,0,0,0,1]

Constraints:

- The number of nodes in each linked list is in the range  $[1, 100]$ .
  - $0 \leq \text{Node.val} \leq 9$
  - It is guaranteed that the list represents a number that does not have leading zeros.
- 

### 3. Longest Substring Without Repeating Characters

## Medium

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

Example 1:

Input: `s = "abcabcbb"`

Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: `s = "bbbbbb"`

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: `s = "pwwkew"`

Output: 3

Explanation: The answer is "wke", with the length of 3. Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

Example 4:

Input: `s = ""`

Output: 0

Constraints:

- `0 <= s.length <= 5 * 104`
  - `s` consists of English letters, digits, symbols and spaces.
- 

## 4. Median of Two Sorted Arrays

### Hard

Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return the median of the two sorted arrays.

The overall run time complexity should be  $O(\log(m+n))$ .

Example 1:

Input: nums1 = [1,3], nums2 = [2]

Output: 2.00000

Explanation: merged array = [1,2,3] and median is 2.

Example 2:

Input: nums1 = [1,2], nums2 = [3,4]

Output: 2.50000

Explanation: merged array = [1,2,3,4] and median is  $(2 + 3) / 2 = 2.5$ .

Example 3:

Input: nums1 = [0,0], nums2 = [0,0]

Output: 0.00000

Example 4:

Input: nums1 = [], nums2 = [1]

Output: 1.00000

Example 5:

Input: nums1 = [2], nums2 = []

Output: 2.00000

Constraints:

- $\text{nums1.length} == m$
- $\text{nums2.length} == n$
- $0 \leq m \leq 1000$
- $0 \leq n \leq 1000$
- $1 \leq m + n \leq 2000$
- $-10^6 \leq \text{nums1}[i], \text{nums2}[i] \leq 10^6$

---

## 5. Longest Palindromic Substring

## Medium

Given a string  $s$ , return *the longest palindromic substring* in  $s$ .

Example 1:

Input:  $s = \text{"babad"}$

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

Example 2:

Input:  $s = \text{"cbbd"}$

Output: "bb"

Example 3:

Input:  $s = \text{"a"}$

Output: "a"

Example 4:

Input:  $s = \text{"ac"}$

Output: "a"

Constraints:

- $1 \leq s.length \leq 1000$
- $s$  consist of only digits and English letters.

---

## 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

Example 3:

Input: s = "A", numRows = 1

Output: "A"

Constraints:

- $1 \leq s.length \leq 1000$
  - s consists of English letters (lower-case and upper-case), ' ', '.' and '..'.
  - $1 \leq \text{numRows} \leq 1000$
- 

## 7. Reverse Integer

Medium

Given a signed 32-bit integer  $x$ , return  $x$  with its digits reversed. If reversing  $x$  causes the value to go outside the signed 32-bit integer range  $[-2^{31}, 2^{31} - 1]$ , then return 0.

Assume the environment does not allow you to store 64-bit integers (signed or unsigned).

Example 1:

Input: x = 123

Output: 321

Example 2:

Input: x = -123

Output: -321

Example 3:

Input: x = 120

Output: 21

Example 4:

Input: x = 0

Output: 0

Constraints:

- $-2^{31} \leq x \leq 2^{31} - 1$
- 

## 8. String to Integer (atoi)

Medium

Implement the `myAtoi(string s)` function, which converts a string to a 32-bit signed integer (similar to C/C++'s `atoi` function).

The algorithm for `myAtoi(string s)` is as follows:

1. Read in and ignore any leading whitespace.
2. Check if the next character (if not already at the end of the string) is '`-`' or '`+`'. Read this character in if it is either. This determines if the final result is negative or positive respectively. Assume the result is positive if neither is present.
3. Read in next the characters until the next non-digit character or the end of the input is reached. The rest of the string is ignored.
4. Convert these digits into an integer (i.e. "`123`"  $\rightarrow$  `123`, "`0032`"  $\rightarrow$  `32`). If no digits were read, then the integer is `0`. Change the sign as necessary (from step 2).
5. If the integer is out of the 32-bit signed integer range  $[-2^{31}, 2^{31} - 1]$ , then clamp the integer so that it remains in the range. Specifically, integers less than  $-2^{31}$  should be clamped to  $-2^{31}$ , and integers greater than  $2^{31} - 1$  should be clamped to  $2^{31} - 1$ .
6. Return the integer as the final result.

Note:

- Only the space character ' ' is considered a whitespace character.
- Do not ignore any characters other than the leading whitespace or the rest of the string after the digits.

Example 1:

Input: s = "42"

Output: 42

Explanation: The underlined characters are what is read in, the caret is the current reader position.

Step 1: "42" (no characters read because there is no leading whitespace)  
  ^

Step 2: "42" (no characters read because there is neither a '-' nor '+')  
  ^

Step 3: "42" ("42" is read in)  
  ^

The parsed integer is 42. Since 42 is in the range [-2<sup>31</sup>, 2<sup>31</sup> - 1], the final result is 42.

Example 2:

Input: s = "-42"

Output: -42

Explanation:

Step 1: " -42" (leading whitespace is read and ignored)  
  ^

Step 2: " -42" ('-' is read, so the result should be negative)  
  ^

Step 3: " -42" ("42" is read in)  
  ^

The parsed integer is -42.

Since -42 is in the range [-2<sup>31</sup>, 2<sup>31</sup> - 1], the final result is -42.

Example 3:

Input: s = "4193 with words"

Output: 4193

### Explanation:

Step 1: "4193 with words" (no characters read because there is no leading whitespace)  
  ^  
Step 2: "4193 with words" (no characters read because there is neither a '-' nor '+')  
  ^  
Step 3: "4193 with words" ("4193" is read in; reading stops because the next character is a non-digit)  
  ^

The parsed integer is 4193.

Since 4193 is in the range [-2<sup>31</sup>, 2<sup>31</sup> - 1], the final result is 4193.

### Example 4:

Input: s = "words and 987"

Output: 0

### Explanation:

Step 1: "words and 987" (no characters read because there is no leading whitespace)  
  ^  
Step 2: "words and 987" (no characters read because there is neither a '-' nor '+')  
  ^  
Step 3: "words and 987" (reading stops immediately because there is a non-digit 'w')  
  ^

The parsed integer is 0 because no digits were read.

Since 0 is in the range [-2<sup>31</sup>, 2<sup>31</sup> - 1], the final result is 0.

### Example 5:

Input: s = "-91283472332"

Output: -2147483648

### Explanation:

Step 1: "-91283472332" (no characters read because there is no leading whitespace)  
  ^  
Step 2: "-91283472332" ('-' is read, so the result should be negative)  
  ^

```
Step 3: "-91283472332" ("91283472332" is read in)
```

```
^
```

```
The parsed integer is -91283472332.
```

Since -91283472332 is less than the lower bound of the range [-2<sup>31</sup>, 2<sup>31</sup> - 1], the final result is clamped to -2<sup>31</sup> = -2147483648.

Constraints:

- $0 \leq s.length \leq 200$
- $s$  consists of English letters (lower-case and upper-case), digits ( $0\text{-}9$ ), ' $'$ ', ' $+$ ', ' $-$ ', and ' $.$ '.

---

## 9. Palindrome Number

Easy

Given an integer  $x$ , return `true` if  $x$  is palindrome integer.

An integer is a palindrome when it reads the same backward as forward. For example, `121` is palindrome while `123` is not.

Example 1:

Input:  $x = 121$

Output: `true`

Example 2:

Input:  $x = -121$

Output: `false`

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

Example 3:

Input:  $x = 10$

Output: `false`

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

Example 4:

Input:  $x = -101$

Output: false

Constraints:

- $-2^{31} \leq x \leq 2^{31} - 1$

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  $'*''$  where:

- $'.'$  Matches any single character.
- $'*''$  Matches zero or more of the preceding element.

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

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

Constraints:

- $1 \leq s.length \leq 20$
- $1 \leq p.length \leq 30$
- s contains only lowercase English letters.
- p contains only lowercase English letters, '.', and '\*'.
- It is guaranteed for each appearance of the character '\*', there will be a previous valid character to match.

---

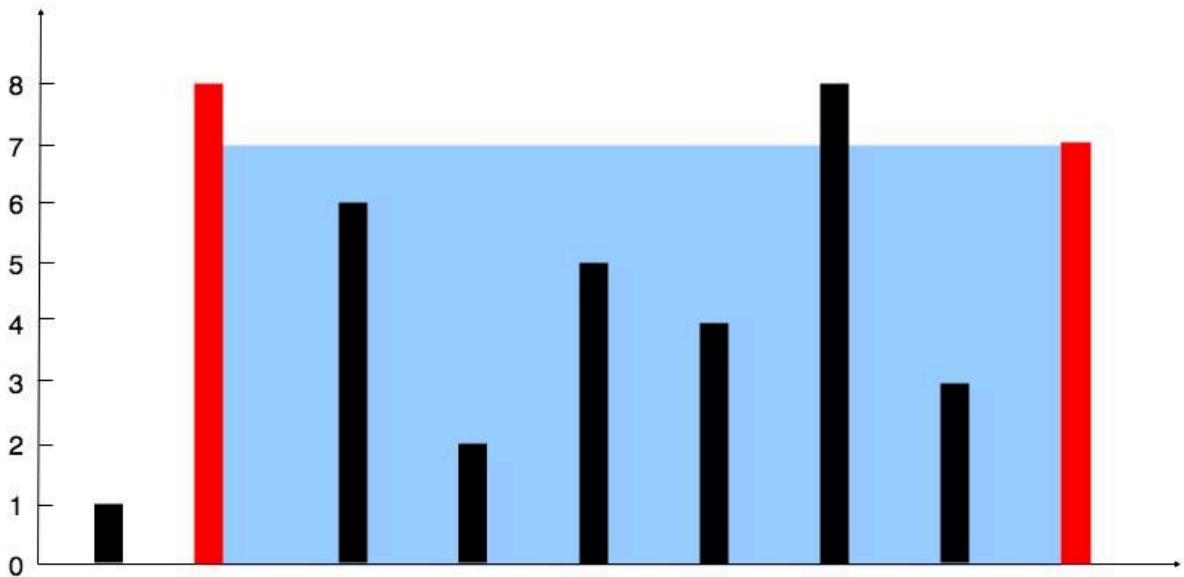
## 11. Container With Most Water

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 the line  $i$  is at  $(i, a_i)$  and  $(i, 0)$ . Find two lines, which, together with the x-axis forms a container, such that the container contains the most water.

Notice that you may not slant the container.

Example 1:



Input: height = [1,8,6,2,5,4,8,3,7]

Output: 49

Explanation: The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container can contain is 49.

Example 2:

Input: height = [1,1]

Output: 1

Example 3:

Input: height = [4,3,2,1,4]

Output: 16

Example 4:

Input: height = [1,2,1]

Output: 2

Constraints:

- $n == \text{height.length}$
- $2 \leq n \leq 105$
- $0 \leq \text{height}[i] \leq 104$

## 12. 3Sum

Medium

Given an integer array `nums`, return all the triplets `[nums[i], nums[j], nums[k]]` such that `i != j, i != k, and j != k, and nums[i] + nums[j] + nums[k] == 0`.

Notice that the solution set must not contain duplicate triplets.

Example 1:

Input: `nums = [-1,0,1,2,-1,-4]`

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

Example 2:

Input: `nums = []`

Output: `[]`

Example 3:

Input: `nums = [0]`

Output: `[]`

Constraints:

- `0 <= nums.length <= 3000`
- `-105 <= nums[i] <= 105`

---

## 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. Return the answer in any order.

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 1:

Input: digits = "23"

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

Example 2:

Input: digits = ""

Output: []

Example 3:

Input: digits = "2"

Output: ["a", "b", "c"]

Constraints:

- $0 \leq \text{digits.length} \leq 4$
- `digits[i]` is a digit in the range `['2', '9']`.

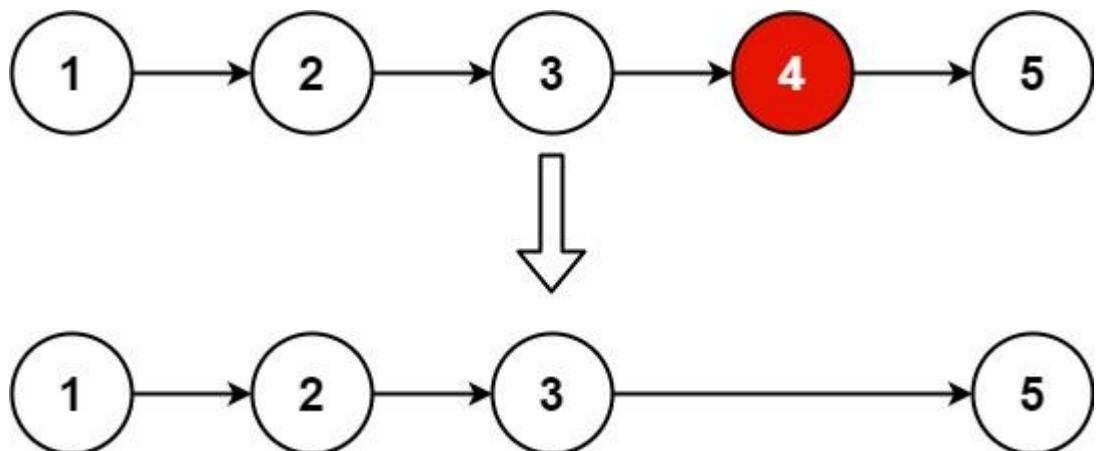
---

## 19. Remove Nth Node From End of List

Medium

Given the `head` of a linked list, remove the `nth` node from the end of the list and return its head.

Example 1:



Input: head = [1,2,3,4,5], n = 2

Output: [1,2,3,5]

Example 2:

Input: head = [1], n = 1

Output: []

Example 3:

Input: head = [1,2], n = 1

Output: [1]

Constraints:

- The number of nodes in the list is `sz`.
- $1 \leq sz \leq 30$
- $0 \leq \text{Node.val} \leq 100$
- $1 \leq n \leq sz$

Follow up: Could you do this in one pass?

## 20. Valid Parentheses

Easy

Given a string `s` 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.

Example 1:

Input: s = "()"

Output: true

Example 2:

Input: s = "()[]{}"

Output: true

Example 3:

Input: s = "(])"

Output: false

Example 4:

Input: s = "([)]"

Output: false

Example 5:

Input: s = "{}[]()

Output: true

Constraints:

- $1 \leq s.length \leq 10^4$
- s consists of parentheses only ' () [] {} '.

---

## 45. Jump Game II

Medium

Given an array of non-negative integers `nums`, you are initially positioned at the first index of the array.

Each element in the array represents your maximum jump length at that position.

Your goal is to reach the last index in the minimum number of jumps.

You can assume that you can always reach the last index.

Example 1:

Input: nums = [2,3,1,1,4]

Output: 2

Explanation: The minimum number of jumps to reach the last index is 2. Jump 1 step from index 0 to 1, then 3 steps to the last index.

Example 2:

Input: nums = [2,3,0,1,4]

Output: 2

Constraints:

- $1 \leq \text{nums.length} \leq 10^4$
  - $0 \leq \text{nums}[i] \leq 1000$
- 

## 39. Combination Sum

Medium

Given an array of distinct integers `candidates` and a target integer `target`, return a *list of all unique combinations of candidates where the chosen numbers sum to target*. You may return the combinations in any order.

The same number may be chosen from `candidates` an unlimited number of times. Two combinations are unique if the frequency of at least one of the chosen numbers is different.

It is guaranteed that the number of unique combinations that sum up to `target` is less than 150 combinations for the given input.

Example 1:

Input: candidates = [2,3,6,7], target = 7

Output: [[2,2,3],[7]]

Explanation:

2 and 3 are candidates, and  $2 + 2 + 3 = 7$ . Note that 2 can be used multiple times.

7 is a candidate, and  $7 = 7$ .

These are the only two combinations.

### Example 2:

Input: candidates = [2,3,5], target = 8

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

### Example 3:

Input: candidates = [2], target = 1

Output: []

### Example 4:

Input: candidates = [1], target = 1

Output: [[1]]

### Example 5:

Input: candidates = [1], target = 2

Output: [[1,1]]

### Constraints:

- $1 \leq \text{candidates.length} \leq 30$
- $1 \leq \text{candidates}[i] \leq 200$
- All elements of candidates are distinct.
- $1 \leq \text{target} \leq 500$