```
//: Playground - noun: a place where people can play

import UIKit

var str = "Hello, playground"

// creating a URL to download
let url = NSURL(string: "https://api.nasa.gov/planetary/apod?
 api_key=DEMO_KEY")! as URL
// creating a URL request for NSURLConnection
let request = URLRequest(url: url)

do{


    let data = try? NSURLConnection.sendSynchronousRequest(request,
     returning: nil)

    let jsonSerialized = try JSONSerialization.jsonObject(with: data!,
     options: []) as? [String:Any]

    if let json = jsonSerialized , let _ = json["url"], let _ =
     json["explanation"]{
    }


}catch{}


// URL for session url
let sessionUrl = URL(string: "https://api.nasa.gov/planetary/apod?
 api_key=DEMO_KEY")

// creating a task with URLSession singleton class
let task = URLSession.shared.dataTask(with: sessionUrl!) { (data, response,
 error) in

    if error != nil {
        print("Error has arised")
    }

    do {
    let jsonSerilized = try JSONSerialization.jsonObject(with: data!,
     options: []) as? [String:Any]

    if let json = jsonSerilized , let url = json["url"], let explanation =
     json["explanation"]{
        print("\n  \(url) \n \n \(explanation)")
        }

    }catch{

    }

}
```

```swift
// resume the past to download
task.resume()

// this need to be run for catching async responses
RunLoop.main.run()

// for finding the some of digits
func findSumOfDigit(num:Int){
    var sum = 0
    var number = num

    while(number != 0 ){
        sum += number % 10
        number = number/10
    }

    print("The sum of all digits of \(num) is \(sum)")
}

findSumOfDigit(num: 9231310)

// to reverse a string
func findReverseOfString(string:String) -> String{
    var reverse = ""
    for char in string{
        reverse = "\(char)" + reverse
    }

    print("\(reverse)")

    return reverse
}

findReverseOfString(string: "Tarun kaushik")

// to find if a string is a palindrome or not
func findPalindrome(string:String){

    let reverseString = findReverseOfString(string: string)

    if reverseString == string{
        print("Its a palindrome")
    }else{
        print("Not a plaindrome")
    }
}


findPalindrome(string: "madam")

// merge sort algo

func merge(left:[Int],right:[Int], Array:[Int]) -> [Int]{
    let lLen = left.count
```

```
    let rLen = right.count

    var A = Array
    //[1,10] adn [1,203]
    var i = 0 ,j = 0,k = 0

    while(i < lLen && j < rLen){

        if left[i] <= right[j]{
            A[k] = left[i]
            i += 1
        }else{
            A[k] = right[j]
            j += 1
        }

        k += 1
    }

    while(i < lLen){
        A[k] = left[i]
        i += 1
        k += 1
    }

    while( j < rLen){
        A[k] = right[j]
        j += 1
        k += 1
    }

    return A

}

func mergeSort(A:[Int]) -> [Int]{

    let len = A.count - 1

    guard len > 0 else{ print("array sorted to single elements \(A)");
     return A}

    let mid = len/2
    let leftArray = Array(A[0...mid])
    let rightArray = Array(A[mid+1 ... len])

    print("left array \(leftArray)")
    print("right arraya \(rightArray)")

    let sortedLeft = mergeSort(A: leftArray)
    let sortedRight = mergeSort(A: rightArray)

    print("sorted left array \(sortedLeft)")
    print("sorted right array  \(sortedRight)")
```

```
    let sortedArray = merge(left: sortedLeft, right: sortedRight, Array: A)

    print(sortedArray)

    return sortedArray
}

let array = [1,10,1,203,1231,2,4,69,30,05,343,5,3,242,5]

mergeSort(A: array)


// selection sort algo
func selectionSort(array:[Int]) -> [Int]{
    var a = array
    let len = array.count - 1
    var i = 0
    while(len != i){
        let searchArray = Array(a[i...len])
        let minIndex = searchMinimum(array: searchArray) + i

        if (minIndex) != i{
            let temp = a[minIndex]
            a[minIndex] = a[i]
            a[i] = temp
        }

        print(a)
        i += 1
    }

    return a
}

func searchMinimum(array:[Int])->Int{
    var minNumber = array[0]

    print(array)
    var minIndex = 0
    var index = 0
    for value in array{
        if minNumber >= value{
            minNumber = value
            minIndex = index
        }
        index += 1
    }

    print(minIndex)
    return minIndex
}

  print(selectionSort(array: array))
```