```swift
var a = 5
var b: Int = 5
let c = 3
var str = "dsa"

var opt: Int?

b += 1

//print(a + (Int(str) ?? 0))


if let number = Int(str) {
//    print(number)
} else {
//    print("nil")
}

func numberize() {
    guard let number = Int(str) else { return }
    print(number)
}

//numberize()

// Arrays and Sets
var arr1: [String] = []
var arr2 = [String]()
var arr3 = Set<String>()

arr1.append("abc")
arr1.append("abc")
arr2.append("ddd")
arr2.append("ccc")

arr1.append(contentsOf: arr2)

//print(arr1.remove(at: 1))
print(arr1)

arr3.insert("abc")
```

```swift
arr3.insert("abc")
arr3.insert("ccc")

print(arr3.sorted())

// Dictionary
var namesOfIntegers: [Int: Any] = [:]
namesOfIntegers[1] = "Abc"
namesOfIntegers[2] = 3

print(namesOfIntegers[1])

for index in 0..<arr1.count {
    print(arr1[index])
}

for _ in 1..<5 {
    print("ads")
}

let sayi = 20
if sayi < 10 {
    print("kucuk")
} else if sayi == 10 {
    print("esit")
} else {
    print("yok")
}

switch sayi {
case 10:
   print("10000")
case 20:
    break
default:
   print("yok")
}

// funcs
func topla(with a: Int, _ b: Int? = nil) -> Int {
    guard let b else { return a }
```

```
    return a+b
}

var toplam: Int = topla(with: 5, 2)
print(toplam)

arr1.forEach { element in
    print(element)
}

// enum
enum Color {
    case red
    case green
    case blue
}

var color: Color = .red

switch color {
case .red:
    print(color)
case .green:
    print(color)
case .blue:
    break
}


// class vs struct
struct Car {
    func start() {
        print("started")
    }

    var name: String
    var model: Int
    var color: Color
}

var car1 = Car(name: "bmw", model: 2010, color: .red)
```

```swift
print(car1.name)

class Vehicle {
    let name: String
    var model: Int
    var color: Color

    init(name: String, model: Int, color: Color) {
        self.name = name
        self.model = model
        self.color = color
    }

    func start() {
        print("vehicle start")
    }
}

class CarClass: Vehicle {
    override func start() {
        print("car start")
    }
}

class Truck: Vehicle {
    var length: Double

    init(name: String, model: Int, color: Color, length: Double) {
        self.length = length
        super.init(name: name, model: model, color: color)
    }

    override func start() {
        super.start()
        print("truck start")
    }

}
```

```swift
var car2 = CarClass(name: "bmw", model: 2000, color: .red)
var car3 = car1
car3.name = "skoda"

var truck: Truck = Truck(name: "volvo", model: 1995, color: .green,
 length: 20.4)
var truck2: Truck?

print("car1: \(car1.name)")
print("car2: \(car2.name)")
print("truck: \(truck.length)")

truck.start()

var aa: Int?
var bb = 4

var total: Int {
    if let aa {
        return aa + bb
    } else {
        return bb
    }
}


print(truck2?.name)
```