```swift
//
//  main.swift
//  CommandLineDemo
//
//  Created by Yinbin Zuo & David Cormier on 2019-01-23.
//

import Foundation

// this stucture implements the random numbers
struct Random {

    static var previousNumber:UInt32?

    static func numberArray(end: Int) -> [Int] {
        var startArr = Array(1...end)
        var resultArr = Array(repeating: 0, count: end)
        for i in 0..<startArr.count {
            let currentCount = UInt32(startArr.count - i)
            let index = Int(arc4random_uniform(currentCount))
            resultArr[i] = startArr[index]
            startArr[index] = startArr[Int(currentCount) - 1]
        }
        return resultArr
    }

    static func numberArray(_ start: Int, end: Int) -> [Int] {
        let scope = end - start
        var startArr = Array(1...scope)
        var resultArr = Array(repeating: 0, count: scope)
        for i in 0..<startArr.count {
            let currentCount = UInt32(startArr.count - i)
            let index = Int(arc4random_uniform(currentCount))
            resultArr[i] = startArr[index]
            startArr[index] = startArr[Int(currentCount) - 1]
        }
        return resultArr.map { $0 + start }
    }

    // this method is for random single number
    static func number(_ count:Int) -> Int {
        var randomNumber = arc4random_uniform(UInt32(count))
        while previousNumber == randomNumber {
            randomNumber = arc4random_uniform(UInt32(count))
        }
        previousNumber = randomNumber
        return Int(randomNumber)
    }
```

```
    // this method is for random single number, that parameters accept
start index and end index
    static func number(_ start:Int, end:Int) -> Int {
        var randomNumber = arc4random_uniform(UInt32(end))
+UInt32(start)
        while previousNumber == randomNumber {
            randomNumber = arc4random_uniform(UInt32(end))
+UInt32(start)
        }
        previousNumber = randomNumber
        return Int(randomNumber)
    }
}

// the structure generate the random puzzle for user
struct Library {
    var words =
["wealth","poor","scream","double","chief","company","bit","seat","cra
ck","whirl","functional","invincible"]

    var currentAnwser: String?

    init() {}

    init(makeByArray:[String]) {
        words = makeByArray
    }

    private func random(withSize size: UInt32) -> Int {
        return Int(arc4random_uniform(size))
    }

    private func random(_ start: UInt32, end: UInt32) -> Int {
        return Int(arc4random_uniform(end) + start)
    }

//    mutating func generatePuzzle() -> String {
//        let covered = random(UInt32(words.min()!.count-2), end:
UInt32(words.max()!.count-2))
//        let selIndex = random(0, end: UInt32(words.count-1))
//        let selWord:String = words[Int(selIndex)]
//        self.currentAnwser = selWord
//        var arr = Array(selWord)
//
//        for index in 0..<covered {
//            let i = random(UInt32(index), end:
UInt32(selWord.count-1))
//            arr[i] = "*"
//        }
```

```
//        return String(arr)
//    }

    mutating func generatePuzzle() -> (String, [Character]) {
        let covered = Random.number(words.min()!.count-2, end:
words.max()!.count-2)
        let selIndex = Random.number(0, end: words.count-1)
        let selWord:String = words[selIndex]
        self.currentAnwser = selWord
        var arr = Array(selWord)

        var currents = Array<Character>()
        for index in 0..<covered {
            let i = Random.number(index, end: selWord.count-1)
            currents.append(arr[i])
            arr[i] = "*"
        }
        return (String(arr),currents) // first one is word which
include the stars, second one is letters array eg. star 1.st** 2.
['a','r']
    }
}

// get first name on user's input
func getFirstName() -> String {
    print("Please enter your first name:")

    let fname = readLine()
    if (fname != nil) {
        let trimed =
fname!.trimmingCharacters(in: .whitespacesAndNewlines)

        if trimed.isEmpty {
            print("Please enter your first name and try again.")
        }
        return fname!
    }
    return ""
}

// get last name on user's input
func getLastName() -> String {
    print("Please enter your last name")
    let lname = readLine()
    if (lname != nil) {
        let trimed =
lname!.trimmingCharacters(in: .whitespacesAndNewlines)

        if trimed.isEmpty {
```

```
            print("Please enter your last name and try again.")
        }else{
            return lname!
        }
    }
    return ""
}

var firstName:String?
var lastName:String?
var option:String?

var lib = Library()

firstName = getFirstName()
lastName = getLastName()


// a while loop for user's options
while option != "3" {
    print("Welcome \(firstName!) \(lastName!)")

    print("Press key to choose the option: 1. Begin the game 2.Modify
your user name 3.Exit")

    option = readLine()

    switch option {
    case "1":
        // move this code in a while loop and implement choose
        let result = lib.generatePuzzle()

        while true {
            print("\(result.0) - Please enter the letters to replace
the 'star' in the word.")

            if let answer = readLine() {
                if answer == String(result.1) {
                    print("Awesome!")
                    print("You win!")
                    break
                }else{
                   print("Please enter your anwser and try again")
                }
            }
        }
        break
    case "2":
        firstName = getFirstName()
```

```
            lastName = getLastName()
            break
        default:
            exit(1)
            break
        }
}
```