```
func fib(_n:Int) -> Int {
    switch _n {
    case 0:
        return 0
        break
        case 1:
            return 1
            break
    default:
        return fib(_n: _n-1) + fib(_n: _n-2)
}
}
print(fib(_n: 7))

func brackets(_s:String) -> Bool {
    var stack:[String] = [String]()
    let hashmap:[String: String] =
        ["[" : "]",
         "{" : "}",
         "(" : ")"
        ]

    for character in _s {
        switch character {
        case "[":
            stack.append("]")
            break
        case "{":
            stack.append("}")
            break
        case "(":
            stack.append(")")
            break
        default:
            if stack.count == 0 { return false }
            for x in hashmap.keys {
                if "\(character)" == hashmap[x] {
                    stack.popLast()
                }
            }
        }
    }
    if stack.count == 0 {
        return true
    }
    return false
}
print(brackets(_s: "()[]{}()()((("))
```