Tavant Programming - June 2024

1. N Queen on a chess board Problem (Hint - backtracking)

The N-Queen problem is to place \(N \) queens on an \(N \times N \) chessboard such that no two queens threaten each other. This means:

1. No two queens can be in the same row.
2. No two queens can be in the same column.
3. No two queens can be on the same diagonal.

The goal is to find all possible configurations of the board where these conditions are satisfied.

2. Problem Statement: Cut the Stick

You are given:
An integer N : the length of a stick.
Three integers x , y , and z : the allowed lengths of each cut.
Your task is to cut the stick into segments such that:
1. The length of each segment is either x , y , or z.
2. The total number of segments is maximized.
If it is not possible to cut the stick in the desired way, return -1.

Input Format:
1. The first line contains an integer N , the length of the stick.
2. The second line contains an integer x , one of the allowed lengths of a cut.
3. The third line contains an integer y , another allowed length of a cut.
4. The fourth line contains an integer z , another allowed length of a cut.

Output Format:
- Print the maximum number of cut segments that can be obtained. If the stick cannot be cut, print \(-1\).

3. How would you implement the Equatable protocol on the Int structure?

Options:

1. static == (rhs: Int) -> Bool
2. static func == (lhs: Int, rhs: Int) -> Bool
3. func equalTo(lhs: Int, rhs: Int) -> Bool
4. static var == (lhs: Int, rhs: Int) -> Bool

4. Condition 0 != 0 is true. How?

Options:

1. func checkEqual<T: Equatable>(value1: T, value2: T) -> Bool { return value1 == value2 }
2. extension Int { static func != (lhs: Int, rhs: Int) -> Bool { return lhs != rhs } }
3. extension Int { static func == (lhs: Int, rhs: Int) -> Bool { return lhs != rhs } }
4. extension Int { static func == (lhs: Int, rhs: Int) -> Bool { return lhs == rhs } }

5. Solve Part A and Part B

```
enum SoftwareTechnology {
   case mobile, desktop, webBackend
}
struct Employee {
   let name: String
   let technology: SoftwareTechnology
   let employeeID: String
  init(name: String, tech: SoftwareTechnology) {
      self.name = name
      self.technology = tech
      self.employeeID = UUID().uuidString
   }
}

class Office {
   private var employees = [Employee]()

   func add(employee: Employee) {
      employees.append(employee)
   }
}

let office = Office()
let rachel = Employee(name: "Rachel S", tech: .mobile)
let hendry = Employee(name: "Henry M", tech: .mobile)
let marc = Employee(name: "Marcus R", tech: .webBackend)

office.add(employee: rachel)
office.add(employee: hendry)
office.add(employee: marc)
```

Part A - An efficient way to remove an employee is:

Pick one
A. Using the filter function to remove a particular element from the collection.
B. By using the index of the element in the collection.
C. By using the map function.
D. Using removeAll functions.

Part B An efficient way to sort employees by their name is:

Pick one
A. By using the sort function in the extension of an array where the element is equal to Employee.
B. By using the sort function in the extension of an array where the index is equal to the employee name index.
C. By using the map function in the extension of an array where the element is equal to the employee name.
D. By inheriting NSObject into the Employee struct

6. Solve Part A and Part B

```
func shuffle<T>(_ array: inout [T]) {
    var length = array.count
    for _ in array {
        let random = arc4random_uniform(UInt32(length))
        if length - 1 != Int(random) {
            array.swapAt(length - 1, Int(random))
        }
        length -= 1
    }
}
var collectionA: [Any] = [1, 2, 4, 6, 87, "Hello", "World", true, 12.04]
collectionA.shuffle()
```

Part A

What is the behavior of the input parameter?

Options:

1. A. The exact behavior of copy on call.
2. B. The exact behavior of copy on write.
3. C. The exact behavior of call by value.
4. D. The exact behavior of call by reference.

Part B

What is the functionality of arc4random_uniform in the above program?

Options:

1. A. It will return an integer value from 0 to 9.
2. B. It will return an integer value from 0 to 8.
3. C. It will return an integer value from 0 to 7.
4. D. It will return an integer value from 1 to 9.