

**Adı:** Onur Hüseyin

**Soyadı:** Çantay

**Öğrenci No:** 152802022

**Adı:** Bertuğ

**Soyadı:** Yılmaz

**Öğrenci No:** 142805009

AMAÇ

- Girilen sayı üzerinden siyah ve beyaz taşlar arasından seçilen belli adette ki taşlardan siyah olanların başta ve yanyana olmayacak şekilde kaç adet dizilim gerçekleştirilebileceğini bulan bir algoritma yazmak.

GIRDILER

- Taş Sayısı dışarıdan girdi olarak alınacaktır.

MANTIK

-Girilen taş miktarı kadar 2'nin üssü olarak alınarak minimum o basamak miktarına ait Binary sistemde sayı bulunur bu sayının 2 ile çarpanı o basamaktaki maksimum sayıya denk gelecektir.bu iki sayı arasında ki döngü ile her bir binary olan sayıyı string e çevirip daha sonrada birer char dizisi haline getirip gerekli kontroller yapıldıktan sonra uygun olanların başka bir string dizisine eklenmesi ile uygun durumlar yazdırılır ve o dizinin Count'u bize uygun dizilim miktarını bize verecektir

## PROGRAM KODU

```
1
2 import Foundation
3
4 func decimalToString(_ deci: Decimal) -> String {
5     let dummy = String(describing: deci)
6     return dummy
7 }
8
9 func stringToInt(_ str: String) -> Int {
10     guard let dummy = Int(str) else {
11         return 0
12     }
13     return dummy
14 }
15
16 func intToBinary(binary : Int) -> String{
17     return String(binary,radix : 2)
18 }
19
20 let blackStone : Character = "0"
21 var lineNumber : Int = 0
22 var aralikAsString : String!
23 var aralikAsInt : Int!
24 var tempCharArray = [Character]()
25 var tempStringArray = [String]()
26 var resultsArray = [String]()
27 var isCorrect : Bool = true
28
29 print("*****\n")
30 print("Sayılar 0 ile başlayamayacağı için siyahları 0 beyazları 1 olarak kabul edilmistir")
31 print("\n*****\n")
32 print("Basamak Sayısını giriniz : ")
33
34 while true {
35     if let response = readline() {
36         if let responseInt = Int(response){
37             if responseInt > 20 {
38                 print("Girdiğiniz Değerin Hesaplanabilmesi Zaman alıyor..")
39                 print("Başka Bir Değer Giriniz...")
40                 continue
41             }
42             lineNumber = responseInt
43             break
44         }else{
45             print("Please enter a valid Value")
46         }
47     }
48 }
49
50 if lineNumber == 0 { //0 lar siyah kabul edildiği için
51     print("hatalı bulunamadı")
52     exit(0)
53 }
54
55 aralikAsString = decimalToString(pow(2, lineNumber)) // 2 Üzeri girilen basamak sayısı kadar olan sayıyı bulur ve string değere dönüştürülür
56 aralikAsInt = stringToInt(aralikAsString) // oluşan bu string değerle işlem yapabilmek için int e çevirildi
57
58 for i in aralikAsInt / 2 ... aralikAsInt - 1 {
59     tempStringArray.removeAll() //farklı değerler için boşaltım gerçekleştirildi
60     tempStringArray.append(intToBinary(binary: i))
61     for item in tempStringArray{
62         tempCharArray = Array(item.characters) //string içerisindeki tüm karakterler char arrayine dönüştürüldü
63
64         for i in 0...tempCharArray.count-1{
65             if tempCharArray.count == i + 1{
66                 break
67             }
68             if tempCharArray[i] == blackStone && tempCharArray[i+1] == blackStone {
69                 isCorrect = false //gerekli şartların sağlanıp sağlanmadığı kontrol edildi
70                 break
71             }else{
72                 isCorrect = true
73             }
74         }
75     }
76 }
```

```

75         if isCorrect {
76             resultsArray.append(item)
77             tempCharArray.removeAll()
78             //char dizisi terkar kullanılabilmek üzere boşaltıldı
79         }
80     }
81 }
82
83 for item in resultsArray {
84     print("Uygun Değerler --> \(item)")
85 }
86
87 print("\n\((lineNumber) lı sayının olasılığı --> \(resultsArray.count)")
88

```

## EKRAN ÇIKTISI

```

*****
Sayılar 0 ile başlayamayacağı için siyahları 0 beyazları 1 olarak kabul edilmistir
*****
Basamak Sayısını giriniz :
15

```

```

Uygun Değerler --> 111111111101101
Uygun Değerler --> 111111111101110
Uygun Değerler --> 111111111101111
Uygun Değerler --> 11111111110101
Uygun Değerler --> 111111111101110
Uygun Değerler --> 111111111101111
Uygun Değerler --> 111111111101010
Uygun Değerler --> 11111111110111
Uygun Değerler --> 1111111111011
Uygun Değerler --> 111111111101
Uygun Değerler --> 11111111110
Uygun Değerler --> 11111111111
15 lı sayının olasılığı --> 987

```

```

Basamak Sayısını giriniz :
''
Please enter a valid Value
asdasd
Please enter a valid Value

Please enter a valid Value
21
Girdiğiniz Değerin Hesaplanabilmesi Zaman alıyor..
Başka Bir Değer Giriniz...

```

## TEST CASE

Risk Level	Yüksek
<b>Purpose</b>	Bu test durumun da kullanıcının gireceği taş miktarı test edilmektedir
<b>Inputs</b>	Girilen taş miktarına bağlı olarak gerekli hesaplamalar yapılması
<b>Pass Criterias</b>	Sonucun doğru şekilde ekrana basılması
<b>Fail Criterias</b>	Basamak Değerinin Tam sayı yerine harf,noktalama işareti veya küsüratlı değer girmesi durumu/aşırı büyük değer girilmesi
<b>Test Procedure</b>	Test Kullanıcısı , Yazılımı desteklenen sistem ve cihazlarda bunlar (Apple Cihazlar-> Macbook) çalıştırarak belirtilen girdilerle testi gerçekleştirmelidir. Test işlemini tamamladıktan sonra test sonucunu Pass /Fail olarak belirterek nedenleri ile raporlamalıdır.