HALİÇ ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BMY549 - İleri Mobil Programlama

1. Dönem 2. Vize Ödevi

Hafta 6 içerisinde bulunan kaynak sonundaki soru

Soru 1: Asal sayıları bulduran programı fonksiyonlar yardımıyla Kotlinde yazınız.

Enes Cevik
23152010013@ogr.halic.edu.tr

İçindekiler

Kaynak Kod

repo: https://github.com/enescevik/IleriMobilOdev2

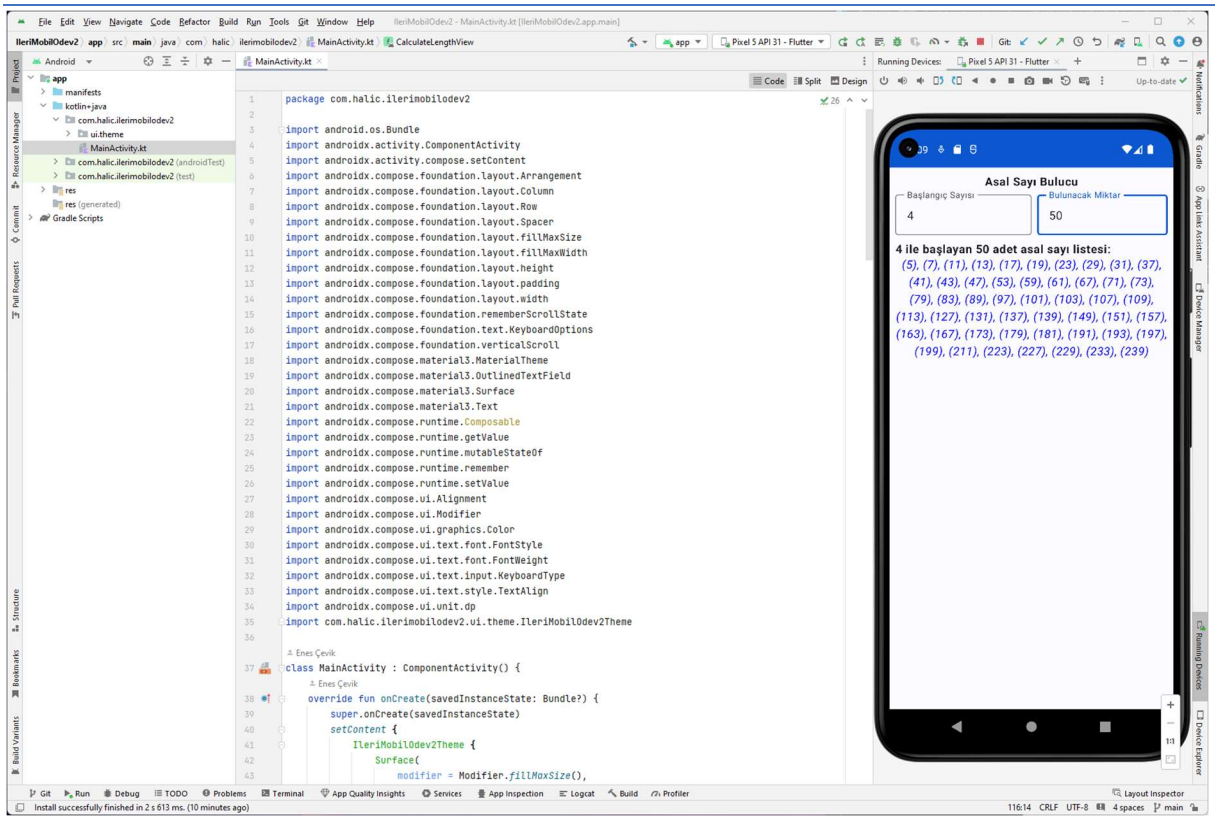
```
package com.halic.ilerimobilodev2

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.foundation.verticalScroll
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontStyle
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import com.halic.ilerimobilodev2.ui.theme.IleriMobilOdev2Theme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            IleriMobilOdev2Theme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Column(
                        modifier = Modifier
                            .fillMaxSize()
                            .padding(8.dp)
                    ) {
                        CalculateLengthView()
                    }
                }
```

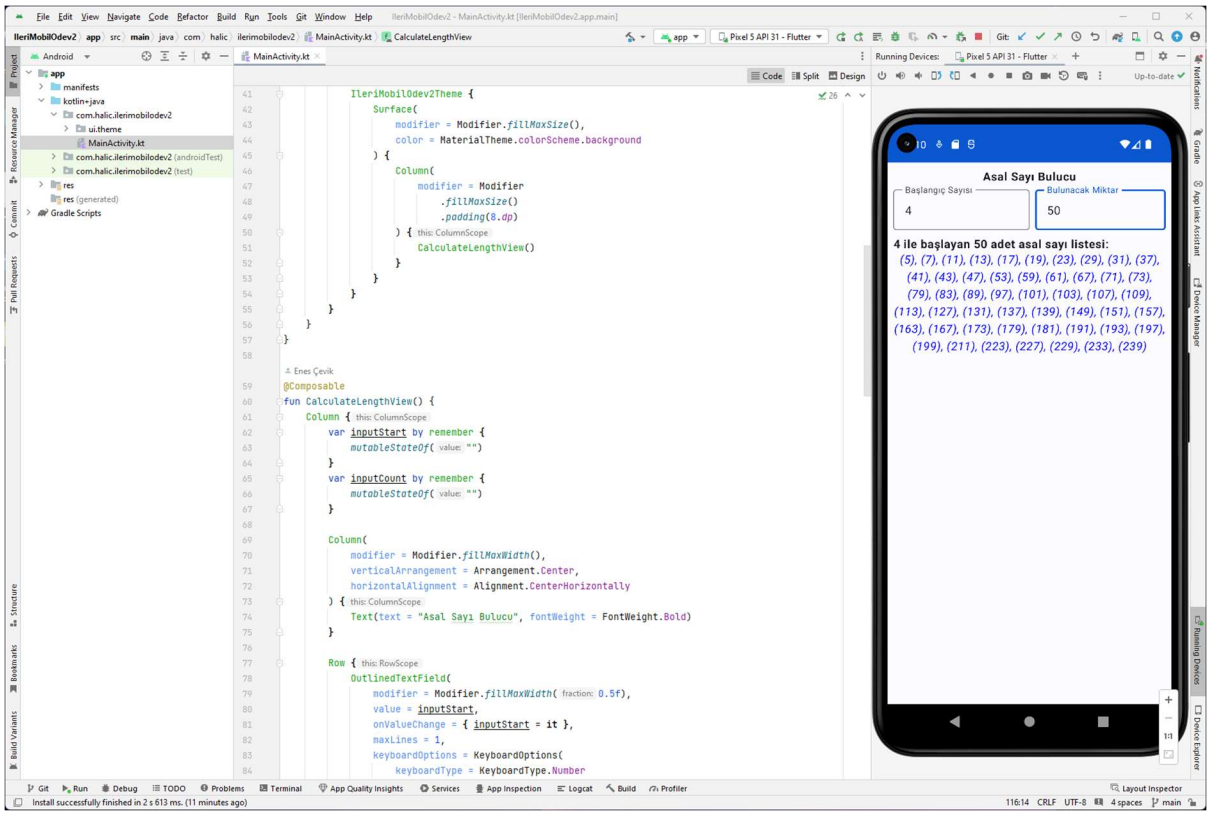
```
            }
        }
    }
}

@Composable
fun CalculateLengthView() {
    Column {
        var inputStart by remember {
            mutableStateOf("")
        }
        var inputCount by remember {
            mutableStateOf("")
        }

        Column(
            modifier = Modifier.fillMaxWidth(),
            verticalArrangement = Arrangement.Center,
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Text(text = "Asal Sayı Bulucu", fontWeight = FontWeight.Bold)
        }

        Row {
            OutlinedTextField(
                modifier = Modifier.fillMaxWidth(0.5f),
                value = inputStart,
                onValueChange = { inputStart = it },
                maxLines = 1,
                keyboardOptions = KeyboardOptions(
                    keyboardType = KeyboardType.Number
                ),
                label = {
                    Text(text = "Başlangıç Sayısı")
                }
            )
            Spacer(modifier = Modifier.width(8.dp))
            OutlinedTextField(
                modifier = Modifier.fillMaxWidth(),
                value = inputCount,
                onValueChange = { inputCount = it },
                maxLines = 1,
                keyboardOptions = KeyboardOptions(
                    keyboardType = KeyboardType.Number
                ),
                label = {
                    Text(text = "Bulunacak Miktar")
                }
            )
        }

        Spacer(modifier = Modifier.height(8.dp))

        val start = inputStart.toIntOrNull()
        val count = inputCount.toIntOrNull()

        if (start == null || count == null) {
            Text(
                text = "Asal sayıları listelemek için başlangıç sayısı ve
bulunacak miktarı giriniz!",
                textAlign = TextAlign.Center,
```

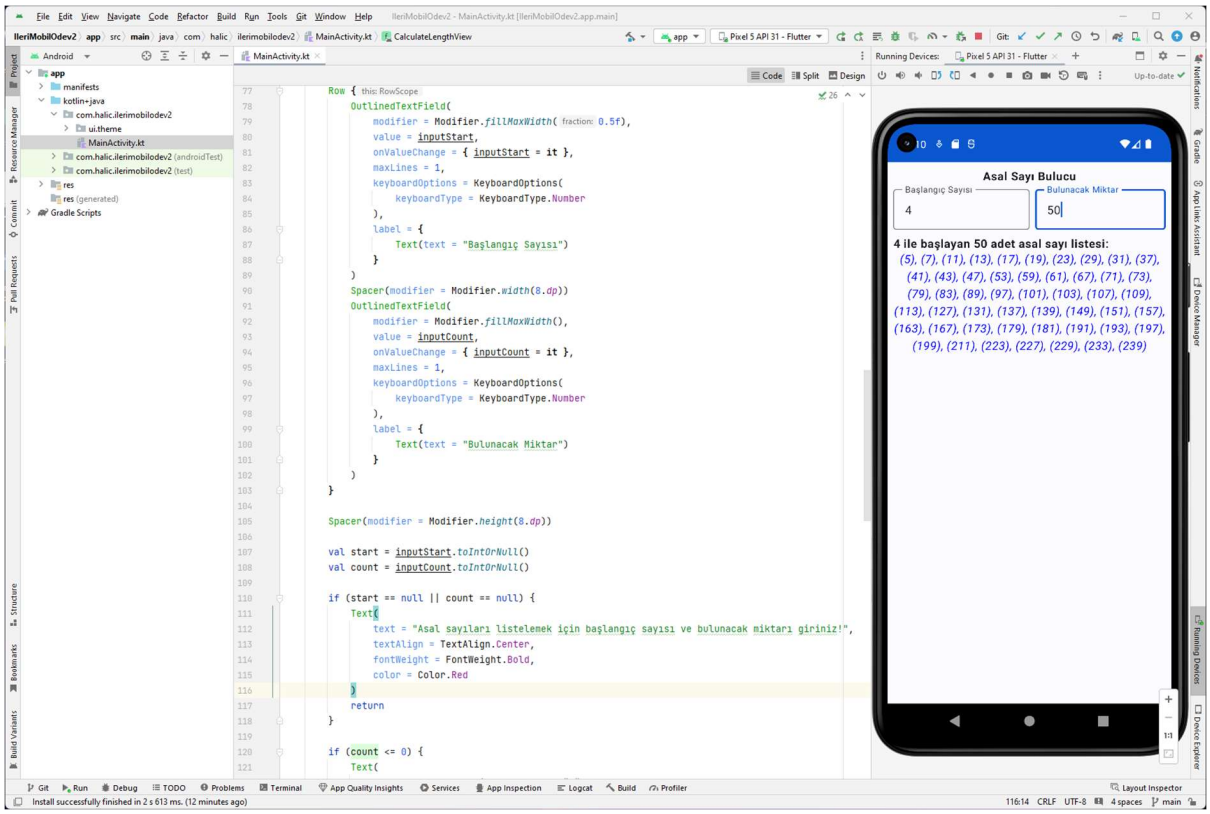
```
                fontWeight = FontWeight.Bold,
                color = Color.Red
            )
            return
        }

        if (count <= 0) {
            Text(
                text = "Bulunacak miktar  0' dan büyük olmalıdır!",
                textAlign = TextAlign.Center,
                fontWeight = FontWeight.Bold,
                color = Color.Red
            )
            return
        }

        if (count > 1000) {
            Text(
                text = "Bulunacak miktar  1,000' den büyük olmamalıdır!",
                textAlign = TextAlign.Center,
                fontWeight = FontWeight.Bold,
                color = Color.Red
            )
            return
        }

        Text(
            text = "%,d ile başlayan %,d adet asal sayı
listesi:".format(start, count),
            fontWeight = FontWeight.Bold
        )

        Column (
            modifier = Modifier
                .verticalScroll(rememberScrollState())
                .weight(weight = 1f, fill = false)
        ) {
            val primeNumbers = calculate(start, count)
            Text(
                text = primeNumbers.joinToString { "(%,d)".format(it) },
                textAlign = TextAlign.Center,
                fontStyle = FontStyle.Italic,
                color = Color.Blue
            )
        }

    }
}

fun calculate(start: Int, count: Int) : MutableList<Int> {
    val result = mutableListOf<Int>()

    var num = start

    do {
        if (isPrime(num)) {
            result.add(num)
        }
        num++
    } while (result.count() < count)
```

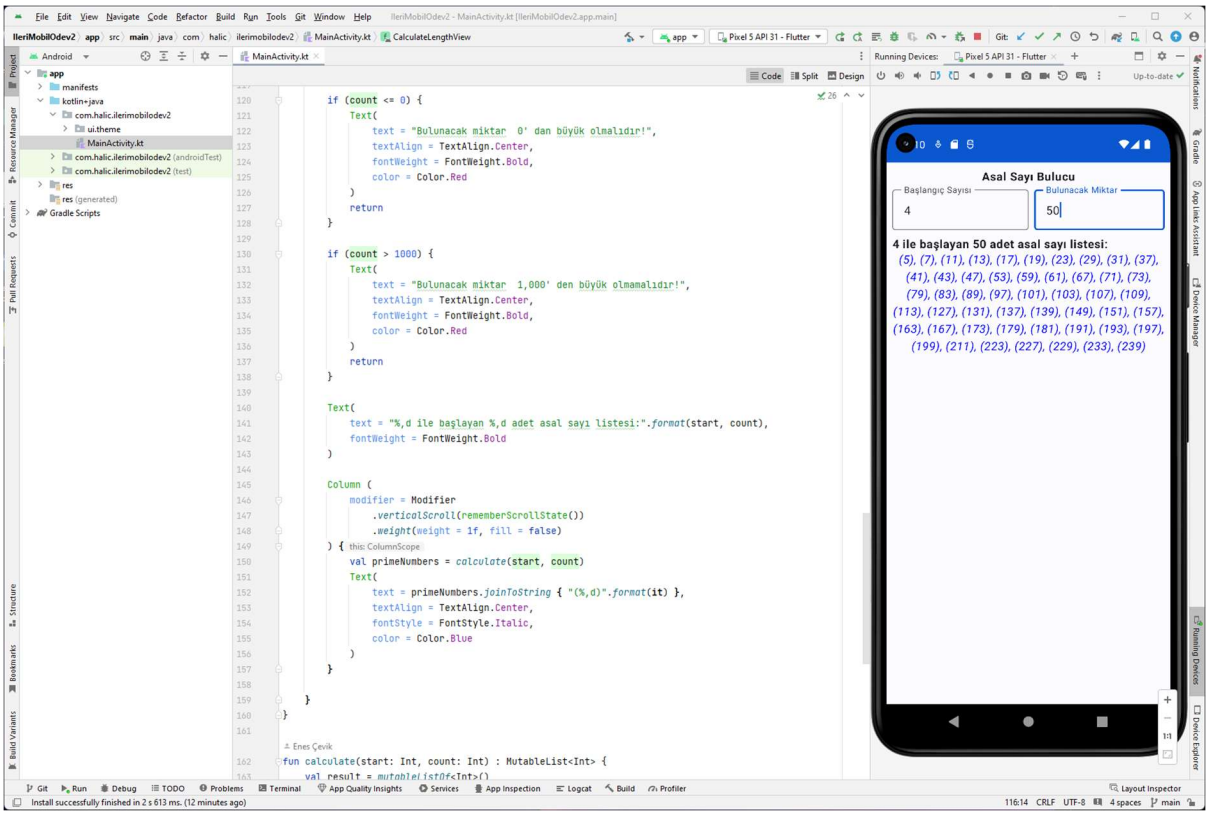
```
    return result
}

fun isPrime(number: Int): Boolean {
    if (number <= 1) {
        return false
    }

    var i = 2
    while (i <= number/i) {
        if (number % i == 0) {
            return false
        }
        i++
    }

    return true
}
```

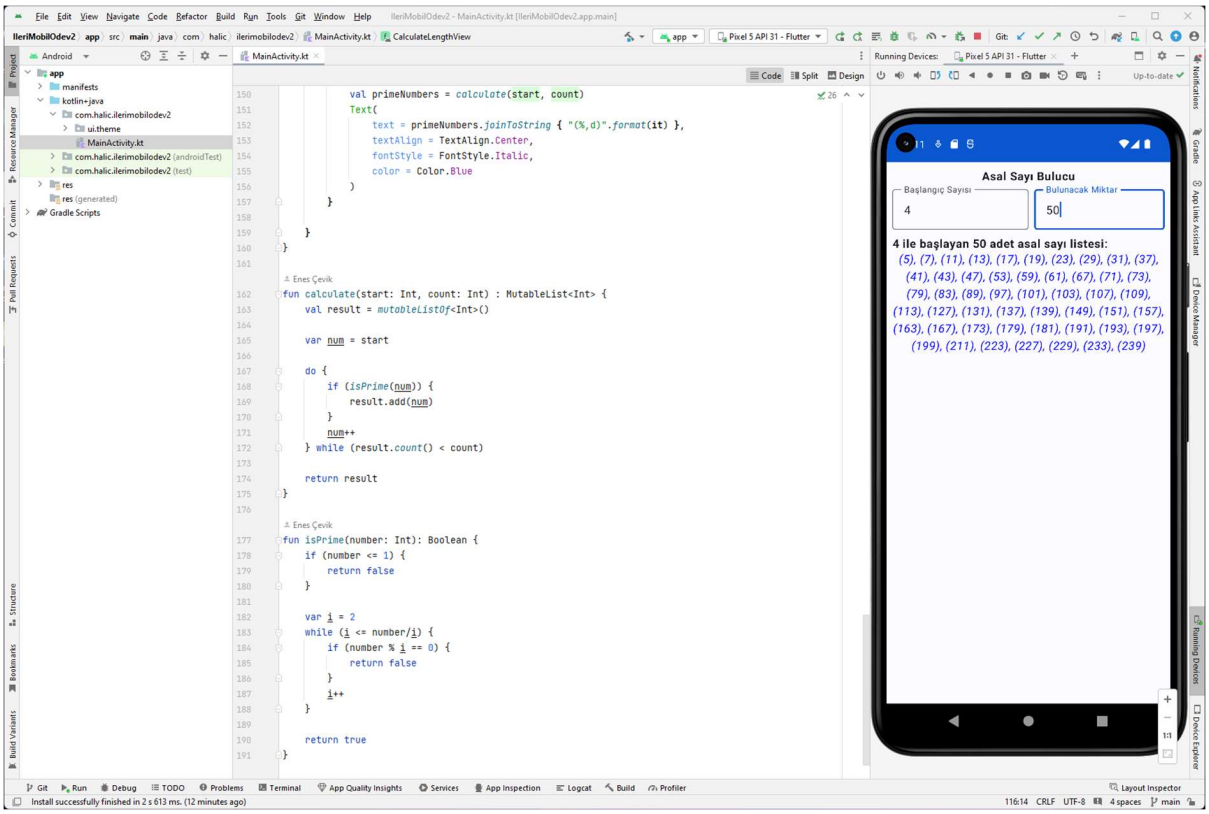
Kodun Ekran Görüntüleri



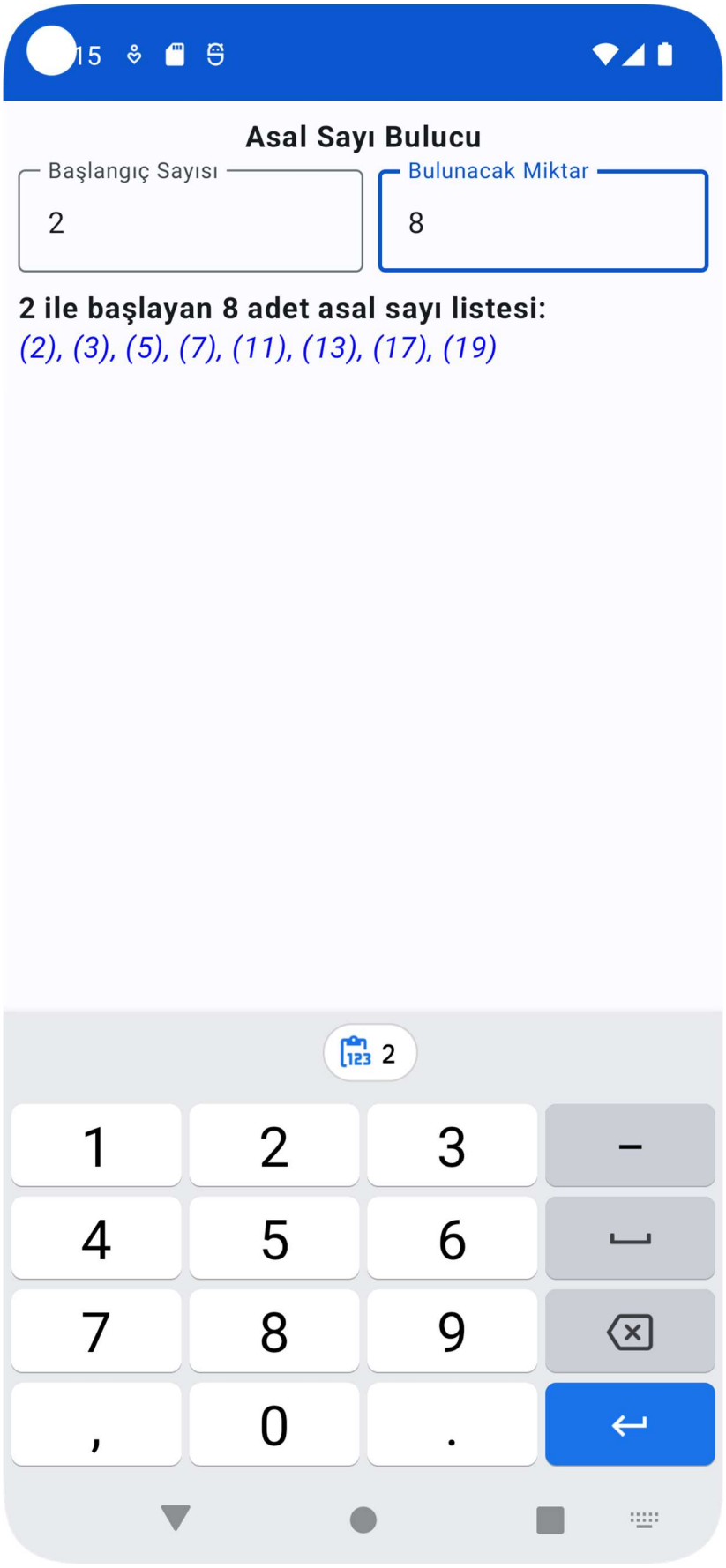

Asal Sayı Bulucu
Başlangıç Sayısı
4
Bulunacak Miktar
50
4 ile başlayan 50 adet asal sayı listesi:
(5), (7), (11), (13), (17), (19), (23), (29), (31), (37), (41), (43), (47), (53), (59), (61), (67), (71), (73), (79), (83), (89), (97), (101), (103), (107), (109), (113), (127), (131), (137), (139), (149), (151), (157), (163), (167), (173), (179), (181), (191), (193), (197), (199), (211), (223), (227), (229), (233), (239)


Asal Sayı Bulucu
Başlangıç Sayısı
4
Bulunacak Miktar
50
4 ile başlayan 50 adet asal sayı listesi:
(5), (7), (11), (13), (17), (19), (23), (29), (31), (37), (41), (43), (47), (53), (59), (61), (67), (71), (73), (79), (83), (89), (97), (101), (103), (107), (109), (113), (127), (131), (137), (139), (149), (151), (157), (163), (167), (173), (179), (181), (191), (193), (197), (199), (211), (223), (227), (229), (233), (239)

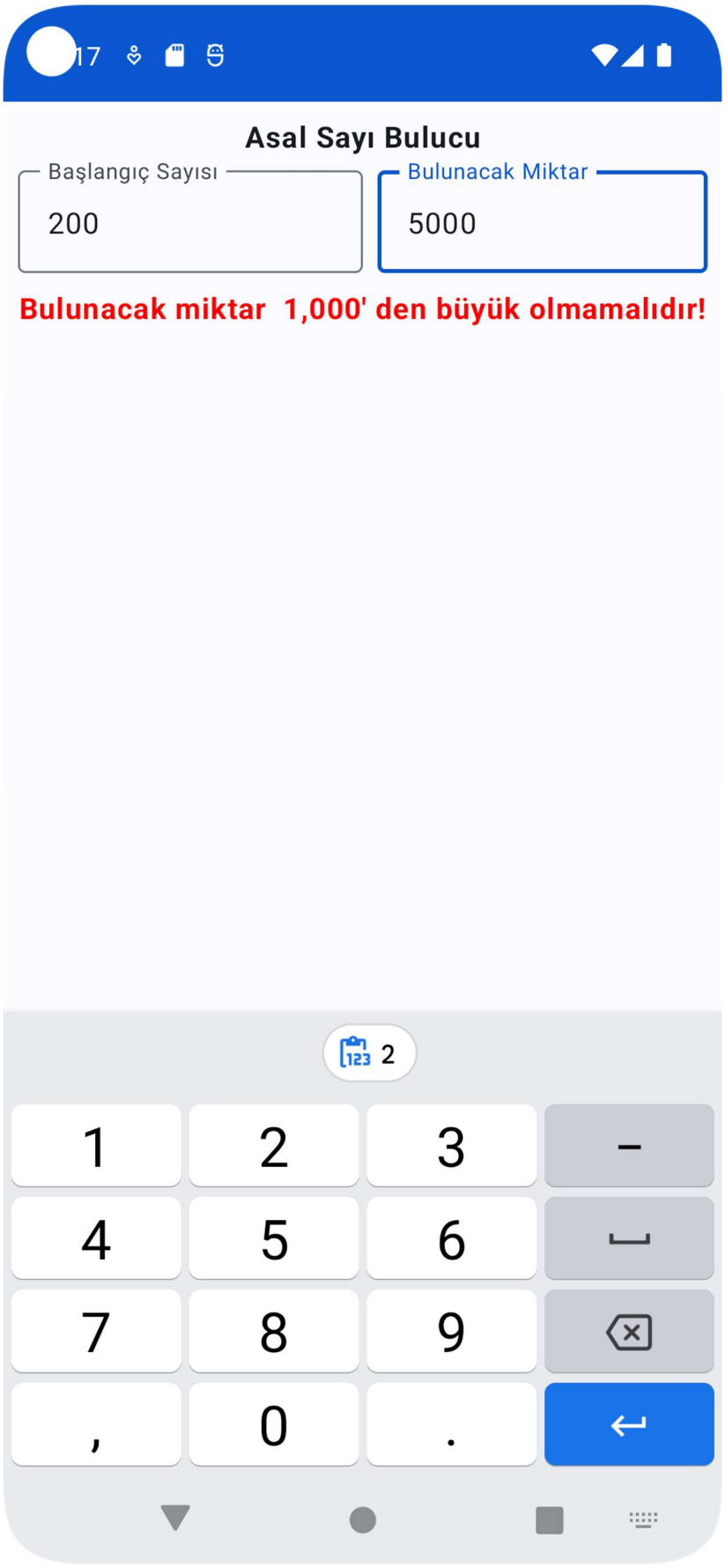

if (count <= 0) {
Text(
text = "Bulunacak miktar `0` dan büyük olmalıdır!",
textAlign = TextAlign.Center,
fontWeight = FontWeight.Bold,
color = Color.Red
)
return
}
if (count > 1000) {
Text(
text = "Bulunacak miktar `1,000` den büyük olmamalıdır!",
textAlign = TextAlign.Center,
fontWeight = FontWeight.Bold,
color = Color.Red
)
return
}
Text(
text = "%,d ile başlayan %,d adet asal sayı listesi:".format(start, count),
fontWeight = FontWeight.Bold
)
Column (
modifier = Modifier
.verticalScroll(rememberScrollState())
.weight(weight = 1f, fill = false)
) { this: ColumnScope
val primeNumbers = calculate(start, count)
Text(
text = primeNumbers.joinToString { "(%,d)".format(it) },
textAlign = TextAlign.Center,
fontStyle = FontStyle.Italic,
color = Color.Blue
)
}
}
}
Enes Çevik
fun calculate(start: Int, count: Int) : MutableList<Int> {
val result = mutableListOf<Int>()
Asal Sayı Bulucu
Başlangıç Sayısı
4
Bulunacak Miktar
50
4 ile başlayan 50 adet asal sayı listesi:
(5), (7), (11), (13), (17), (19), (23), (29), (31), (37), (41), (43), (47), (53), (59), (61), (67), (71), (73), (79), (83), (89), (97), (101), (103), (107), (109), (113), (127), (131), (137), (139), (149), (151), (157), (163), (167), (173), (179), (181), (191), (193), (197), (199), (211), (223), (227), (229), (233), (239)

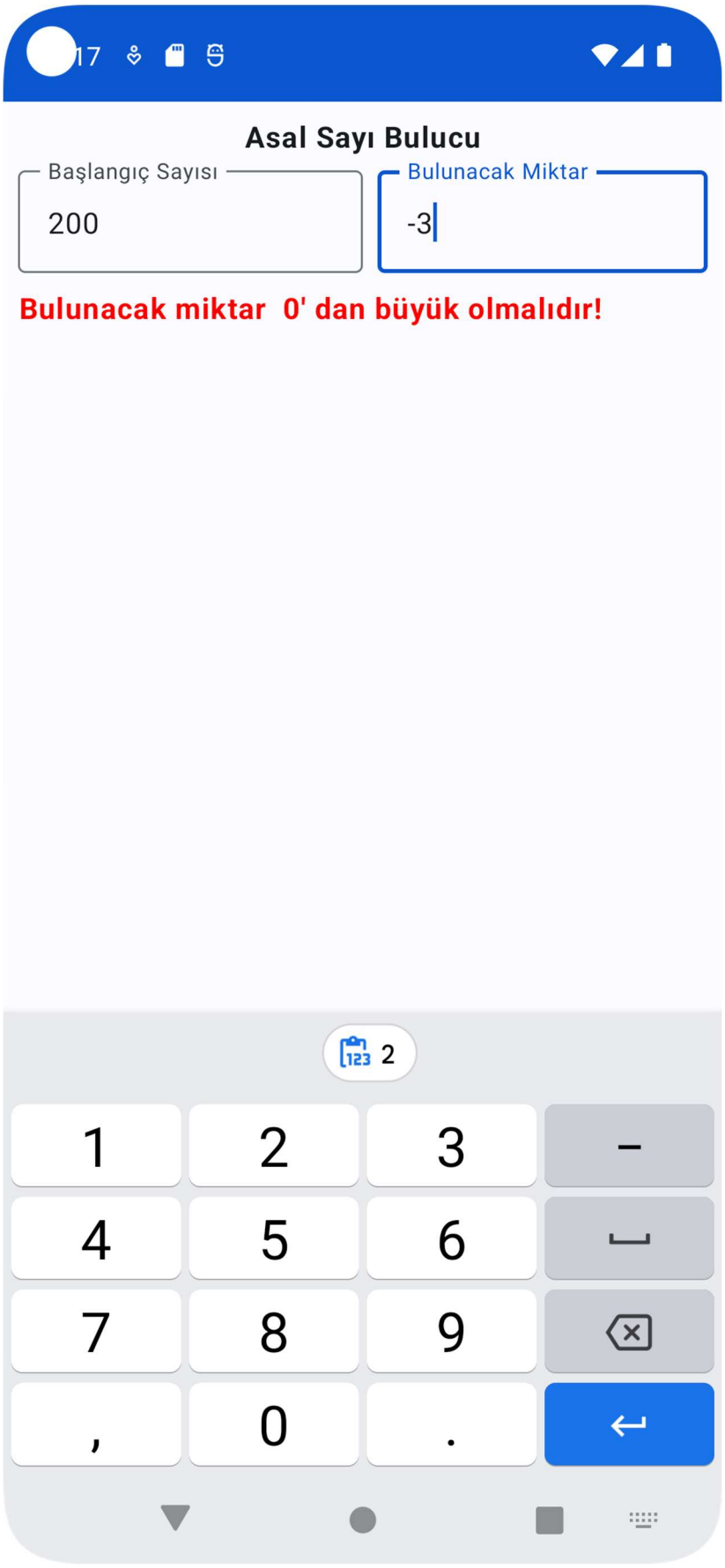

val primeNumbers = calculate(start, count)
Text(
text = primeNumbers.joinToString { "(%,d)".format(it) },
textAlign = TextAlign.Center,
fontStyle = FontStyle.Italic,
color = Color.Blue
)
}
}
}
Enes Çevik
fun calculate(start: Int, count: Int) : MutableList<Int> {
val result = mutableListOf<Int>()
var num = start
do {
if (isPrime(num)) {
result.add(num)
}
num++
} while (result.count() < count)
return result
}
Enes Çevik
fun isPrime(number: Int): Boolean {
if (number <= 1) {
return false
}
var i = 2
while (i <= number/i) {
if (number % i == 0) {
return false
}
i++
}
return true
}
Asal Sayı Bulucu
Başlangıç Sayısı
4
Bulunacak Miktar
50
4 ile başlayan 50 adet asal sayı listesi:
(5), (7), (11), (13), (17), (19), (23), (29), (31), (37), (41), (43), (47), (53), (59), (61), (67), (71), (73), (79), (83), (89), (97), (101), (103), (107), (109), (113), (127), (131), (137), (139), (149), (151), (157), (163), (167), (173), (179), (181), (191), (193), (197), (199), (211), (223), (227), (229), (233), (239)

Uygulama Ekran Görüntüleri


15
Asal Sayı Bulucu
Başlangıç Sayısı
2
Bulunacak Miktar
8
2 ile başlayan 8 adet asal sayı listesi:
(2), (3), (5), (7), (11), (13), (17), (19)
123 2
1
2
3
4
5
6
7
8
9
,
0
.


Asal Sayı Bulucu
Başlangıç Sayısı
200
Bulunacak Miktar
500
200 ile başlayan 500 adet asal sayı listesi:
(211), (223), (227), (229), (233), (239), (241), (251),
(257), (263), (269), (271), (277), (281), (283), (293),
(307), (311), (313), (317), (331), (337), (347), (349),
(353), (359), (367), (373), (379), (383), (389), (397),
(401), (409), (419), (421), (431), (433), (439), (443),
(449), (457), (461), (463), (467), (479), (487), (491),
(499), (503), (509), (521), (523), (541), (547), (557),
(563), (569), (571), (577), (587), (593), (599), (601),
(607), (613), (617), (619), (631), (641), (643), (647),
(653), (659), (661), (673), (677), (683), (691), (701),
(709), (719), (727), (733), (739), (743), (751), (757),
(761), (769), (773), (787), (797), (809), (811), (821),
(823), (827), (829), (839), (853), (857), (859), (863),
(877), (881), (883), (887), (907), (911), (919), (929),
(937), (941), (947), (953), (967), (971), (977), (983),
(991) (997) (1 009) (1 013) (1 019) (1 021)
2
1
2
3
–
4
5
6
7
8
9
,
0
.


17
Asal Sayı Bulucu
Başlangıç Sayısı
200
Bulunacak Miktar
5000
Bulunacak miktar 1,000' den büyük olmamalıdır!
2
1
2
3
4
5
6
7
8
9
,
0
.


17
Asal Sayı Bulucu
Başlangıç Sayısı
200
Bulunacak Miktar
-3
Bulunacak miktar 0' dan büyük olmalıdır!
2
1
2
3
–
4
5
6
7
8
9
,
0
.