PRACTICAL FILE

Flutter and Dart Lab.

(MCA-267)

Submitted by:
Tanish Sharma
70311104422
MCA 3rd sem

Submitted to:
Dr. Sushma Bahuguna
(Sr. Asst. Professor)

BANARSIDAS CHANDIWALA INSTITUTE OF INFORMATION TECHNOLOGY *AFFILIATED WITH*

GURU GOBIND SINGH INDRAPRASTH UNIVERSITY, DELHI

(2022 -2024)

INDEX

Sr.No	Title	Pg.No	Sign
1.	Write a Dart Program to convert Temperature to and from Celsius Fahrenheit	4	
2.	Write a Dart program to check a pair of numbers and return true if one of the numbers is 50 or if their sum is 50.	5	
3.	Write a program to print Sum of Digits of a number.	6	
4.	Write a JavaScript program to check whether a given positive number is divisible by 2 to 11.	7	
5.	Make a list of factors of a given number. The actual factors of 108 which are 1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54, and 108.	10	
6.	Make a list of all prime factors of a given number. The prime factorization of the number 108 gives us 108 = 2 × 2 × 3 × 3 × 3	11	
7.	Finding the Number of Factors of given number. The number of factors of 108 is 12.	12	
8.	Write a program to reverse the numbers given in a list	13	
9.	Write a Dart program to sort two given Lists of integers, merge and create another sorted array.	14	
10.	Create a class Mobile, declare fields for mobile specs (i.e brand,color, camera) and initialized constructor and create three objects initialize and print details.	16	
11.	Write a program to print number into words.	17	
12.	Write a program to print the number into words.	19	
13.	Write a program to print binary equivalent of a number.	21	
14.	Write a program to print the given patterns.	22	
15.	Write a program to create the Application to perform the following operations on the number: Increment, Decrement, Square, Reset Check Prime, Check Even/Odd, Check Palindrome, Compute Factorial	27	
16.	Create a flutter application using Bottom Navigation Bar, Bottom app Bar and Tab Bar.	33	
17.	Create a flutter application that obtains two integers from the user and prints their product, difference and quotient (division).	37	
18.	Create a flutter that converts a number from Fahrenheit to degree and vice-versa.	40	

19.	Create a flutter application with UI to display the image of the image URL of TextField.	43	
20.	Create a flutter application using Inkwell , OnTap event widgets.	45	
21.	Create a flutter application using Draggable, Dragtarget widgets.	49	
22.	Create a Tic Tac Toe Application.	52	
23.	Create a flutter application to using stack widget.	57	
24.	Design a login Page and navigate to the next page if correct credentials are entered	59	
25.	Create a flutter application to play and pause a video.	64	
26.	Create the list of students and display using list view.	67	
27.	Create a flutter app using Hero Widget	71	
28.	Create a flutter app using Card Widget	74	
29.	Write a dart program to read and write data to a JSON file	77	
30.	Create a flutter app using Grid view widget and hero widget	79	

Q1. Write a Dart Program to convert Temperature to and from Celsius-Fahrenheit.

```
import 'dart:io';

double celsiusToFahrenheit(double celsius) {
  return (celsius * 9/5) + 32;
}

double fahrenheitToCelsius(double fahrenheit) {
  return (fahrenheit - 32) * 5/9;
}

void main() {
  print("Choose conversion:");
  print("1. Celsius to Fahrenheit");
  print("2. Fahrenheit to Celsius");

  int choice = int.parse(stdin.readLineSync() ?? '');

  if (choice == 1) {
    print("Enter temperature in Celsius: ");
    double celsius = double.parse(stdin.readLineSync() ?? '');
    double fahrenheit = celsiusToFahrenheit(celsius);
    print("$celsius Celsius is equal to $fahrenheit Fahrenheit.");
  } else if (choice == 2) {
    print("Enter temperature in Fahrenheit: ");
    double fahrenheit = double.parse(stdin.readLineSync() ?? '');
    double celsius = fahrenheitToCelsius(fahrenheit);
    print("$fahrenheit Fahrenheit is equal to $celsius Celsius.");
  } else {
    print("Invalid choice.");
  }
}
```

Output:

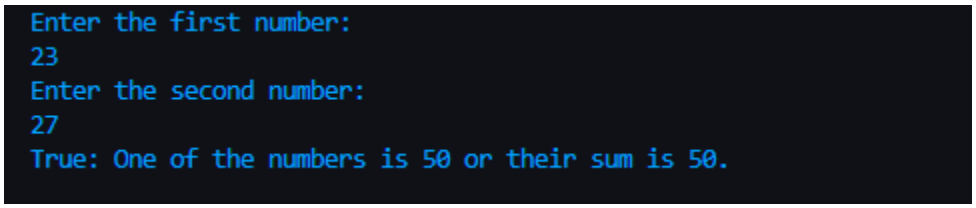
```
Choose conversion:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
1
Enter temperature in Celsius:
35
35.0 Celsius is equal to 95.0 Fahrenheit.
```

```
Choose conversion:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
2
Enter temperature in Fahrenheit:
43
43.0 Fahrenheit is equal to 6.111111111111111 Celsius.
```

Q2. Write a Dart program to check a pair of numbers and return true if one of the numbers is 50 or if their sum is 50.

```
import 'dart:io';

bool checkNumberPair(int num1, int num2) {
  return num1 == 50 || num2 == 50 || num1 + num2 == 50;
}

void main() {
  print("Enter the first number: ");
  int number1 = int.parse(stdin.readLineSync() ?? '');

  print("Enter the second number: ");
  int number2 = int.parse(stdin.readLineSync() ?? '');

  bool result = checkNumberPair(number1, number2);

  if (result) {
    print("True: One of the numbers is 50 or their sum is 50.");
  } else {
    print("False: Neither of the numbers is 50, and their sum is not 50.");
  }
}
```

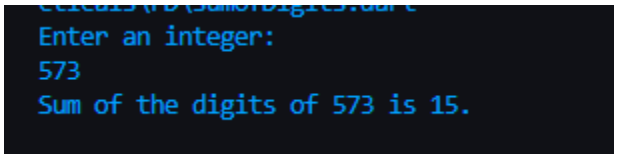
Output:



Q3. Write a program to print Sum of Digits of a number.

```
Import ‘dart:io’;

int calculateSumOfDigits(int number) {
  int sum = 0;
  while (number > 0) {
    sum += number % 10;
    number ~/= 10; //performs int division
  }
  return sum;
}

void main() {
  print(“Enter an integer: “);
  int num = int.parse(stdin.readLineSync() ?? ‘’);

  int sum = calculateSumOfDigits(num);

  print(“Sum of the digits of $num is $sum.”);
}
```

Output:

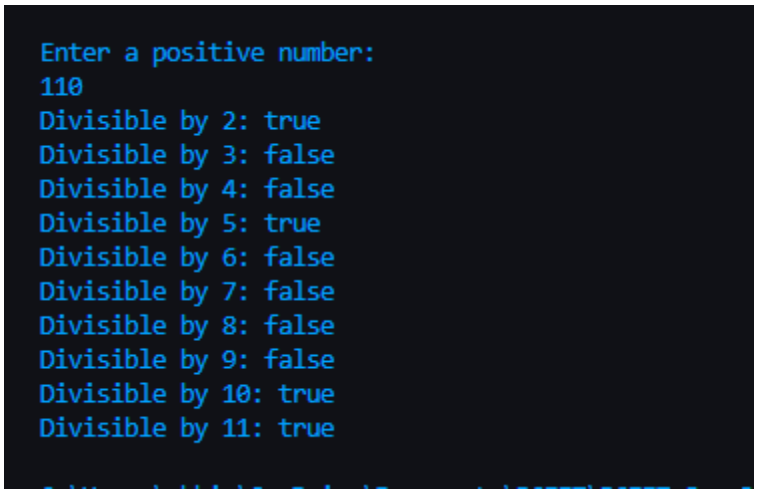


Q4.Write a program to check whether a given positive number is divisible by 2 to 11.

```
import 'dart:io';

bool isDivisibleBy2(int number) {
  return number % 2 == 0;
}

bool isDivisibleBy3(int number) {
  return sumOfDigits(number) % 3 == 0;
}

bool isDivisibleBy4(int number) {
  final lastTwoDigits = number % 100;
  return lastTwoDigits % 4 == 0;
}

bool isDivisibleBy5(int number) {
  return number % 10 == 0 || number % 10 == 5;
}

bool isDivisibleBy6(int number) {
  return isDivisibleBy2(number) && isDivisibleBy3(number);
}

bool isDivisibleBy7(int number) {
  final remainingPart = (number ~/ 10);
  final difference = 2 * (number % 10) - remainingPart;
  return difference % 7 == 0;
}

bool isDivisibleBy8(int number) {
  final lastThreeDigits = number % 1000;
  return lastThreeDigits % 8 == 0;
}

bool isDivisibleBy9(int number) {
  return sumOfDigits(number) % 9 == 0;
}

bool isDivisibleBy10(int number) {
  return number % 10 == 0;
}

bool isDivisibleBy11(int number) {
  List<int> digits = [];
  while (number > 0) {
    digits.add(number % 10);
    number ~/= 10;
```

```
  }

  int sumOdd = 0;
  int sumEven = 0;

  for (int i = 0; i < digits.length; i++) {
    if (i % 2 == 0) {
      sumOdd += digits[i];
    } else {
      sumEven += digits[i];
    }
  }

  int difference = sumOdd - sumEven;

  return difference % 11 == 0;
}


int sumOfDigits(int number) {
  var sum = 0;
  while (number > 0) {
    sum += number % 10;
    number ~/= 10;
  }
  return sum;
}

void main() {
  print("Enter a positive number:");
  int N = int.parse(stdin.readLineSync() ?? '');

  print("Divisible by 2: ${isDivisibleBy2(N)}");
  print("Divisible by 3: ${isDivisibleBy3(N)}");
  print("Divisible by 4: ${isDivisibleBy4(N)}");
  print("Divisible by 5: ${isDivisibleBy5(N)}");
  print("Divisible by 6: ${isDivisibleBy6(N)}");
  print("Divisible by 7: ${isDivisibleBy7(N)}");
  print("Divisible by 8: ${isDivisibleBy8(N)}");
  print("Divisible by 9: ${isDivisibleBy9(N)}");
  print("Divisible by 10: ${isDivisibleBy10(N)}");
  print("Divisible by 11: ${isDivisibleBy11(N)}");
}
```

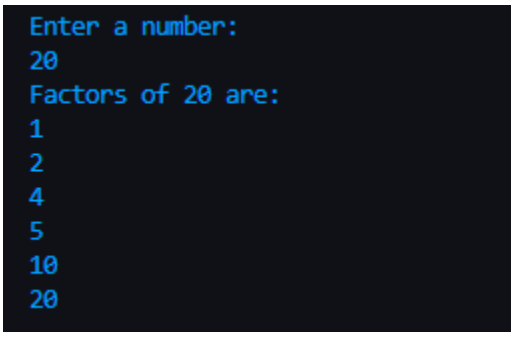
Output


Enter a positive number:
110
Divisible by 2: true
Divisible by 3: false
Divisible by 4: false
Divisible by 5: true
Divisible by 6: false
Divisible by 7: false
Divisible by 8: false
Divisible by 9: false
Divisible by 10: true
Divisible by 11: true

Q5. Make a list of factors of a given number. The actual factors of 108 which are 1, 2, 3, 4, 6, 9, 12, 18, 27, 36, 54, and 108.

```
import 'dart:io';

List<int> findFactors(int number) {
  List<int> factors = [];

  for (int i = 1; i <= number; i++) {
    if (number % i == 0) {
    factors.add(i);
    }
  }

  return factors;
}

void main() {
  print("Enter a number:");
  int number = int.parse(stdin.readLineSync() ?? '');

  List<int> factors = findFactors(number);

  print("Factors of $number are:");
  for (int factor in factors) {
  print(factor);
  }
}
```

Output:



Q6. Make a list of all prime factors of a given number. The prime factorization of the number 108 gives us 108 = 2 × 2 × 3 × 3 × 3

```
import ‘dart:io’;

void main() {
  print(“Enter a number:”);
  int number = int.parse(stdin.readLineSync() ?? ‘’);

  List<int> primeFactors = findPrimeFactors(number);

  if (primeFactors.isEmpty) {
    print(“The number $number has no prime factors.”);
  } else {
    print(“Prime factors of $number are: $primeFactors”);
  }
}

List<int> findPrimeFactors(int number) {
  List<int> primeFactors = [];
  int divisor = 2;

  while (number > 1) {
    while (number % divisor == 0) {
      primeFactors.add(divisor);
      number = number ~/ divisor;
    }
    divisor++;
  }

  return primeFactors;
}
```

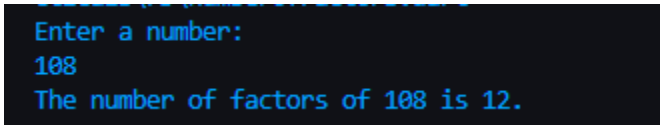
Output:

```
Enter a number:
15
Prime factors of 15 are: [3, 5]
```

Q7. Finding the Number of Factors of given number. The number of factors of 108 is 12.

```
import 'dart:io';

void main() {
  print("Enter a number:");
  int number = int.parse(stdin.readLineSync() ?? '');

  int numberOfFactors = countFactors(number);

  print("The number of factors of $number is $numberOfFactors.");
}

int countFactors(int number) {
  int count = 0;

  for (int i = 1; i <= number; i++) {
    if (number % i == 0) {
    count++;
    }
  }

  return count;
}
```

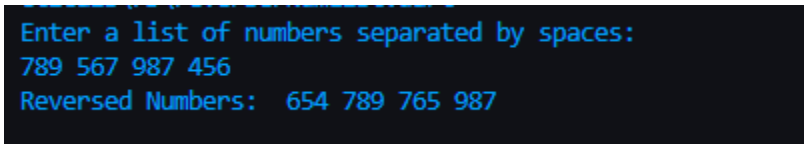
Output:



Q8. Write a program to reverse the numbers given in a list.

```
import 'dart:io';

void main() {
  // Take input from the user
  print("Enter a list of numbers separated by spaces:");
  String input = stdin.readLineSync() ?? "";

  // Split the input string into a list of strings
  List<String> numbers = input.split(" ");

  // Reverse each number in the list and then reverse the entire list
  List<String> reversedNumbers = reverseList(numbers);

  // Print the reversed list
  print("Reversed Numbers: ${reversedNumbers.join(' ')}");
}

// Function to reverse each number in the list and then reverse the entire list
List<String> reverseList(List<String> numbers) {
  List<String> reversedNumbers = [];

  for (String number in numbers) {
    List<String> digits = number.split('');
    digits = digits.reversed.toList();
    reversedNumbers.add(digits.join(''));
  }

  return reversedNumbers.reversed.toList();
}
```

Output:



Q9. Write a Dart program to sort two given Lists of integers, merge and create another sorted array.

```
import 'dart:io';

void main() {
  List<int> array1 = [];
  List<int> array2 = [];

  // Input for the first list
  print("Enter the elements for the first list (separated by spaces):");
  String input1 = stdin.readLineSync() ?? '';
  array1 = input1.split(' ').map((str) => int.tryParse(str) ?? 0).toList();

  // Input for the second list
  print("Enter the elements for the second list (separated by spaces):");
  String input2 = stdin.readLineSync() ?? '';
  array2 = input2.split(' ').map((str) => int.tryParse(str) ?? 0).toList();

  // Manually sort both arrays
  array1 = manualSort(array1);
  array2 = manualSort(array2);

  // Merge the sorted arrays
  List<int> result = mergeSortedArrays(array1, array2);

  print("Merged and Sorted Result: $result");
}

List<int> manualSort(List<int> list) {
  for (int i = 0; i < list.length - 1; i++) {
  for (int j = i + 1; j < list.length; j++) {
      if (list[i] > list[j]) {
        int temp = list[i];
        list[i] = list[j];
        list[j] = temp;
      }
    }
  }
  return list;
}

List<int> mergeSortedArrays(List<int> array1, List<int> array2) {
  List<int> result = [];

  int i = 0;
  int j = 0;

  while (i < array1.length && j < array2.length) {
    if (array1[i] < array2[j]) {
    result.add(array1[i]);
```

```
    i++;
   } else {
    result.add(array2[j]);
    j++;
   }
  }

  // Add any remaining elements from both arrays
  while (i < array1.length) {
   result.add(array1[i]);
   i++;
  }

  while (j < array2.length) {
   result.add(array2[j]);
   j++;
  }

  return result;
}
```

Output:

```
Enter the elements for the first list (separated by spaces):
3 9 1
Enter the elements for the second list (separated by spaces):
2 6 -4
Merged and Sorted Result: [-4, 1, 2, 3, 6, 9]
```

Q10. Create a class Mobile, declare fields for mobile specs (i.e brand,color, camera) and initialized constructor and create three objects initialize and print details.

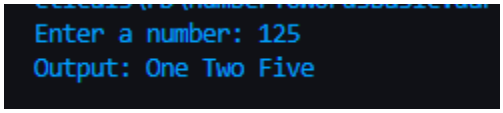
```
class Mobile {
  String brand;
  String color;
  double camera;

  // Constructor to initialize the mobile object
  Mobile(this.brand, this.color, this.camera);

  // Method to print mobile details
  void printDetails() {
  print("Brand: $brand");
  print("Color: $color");
  print("Camera: $camera MP");
  }
}

void main() {
  // Initialize three mobile objects
  Mobile mobile1 = Mobile("Samsung", "Black", 12.0);
  Mobile mobile2 = Mobile("iPhone", "White", 16.0);
  Mobile mobile3 = Mobile("Google Pixel", "Silver", 12.0);

  // Print details of the mobile objects
  print("Mobile 1 Details:");
  mobile1.printDetails();

  print("\nMobile 2 Details:");
  mobile2.printDetails();

  print("\nMobile 3 Details:");
  mobile3.printDetails();
}
```

Q11. Write a program to print a number into words.

```
import 'dart:io';

void main() {
  // Take input from the user
  stdout.write("Enter a number: ");
  int number = int.parse(stdin.readLineSync()!);

  // Convert the number into words
  String result = convertNumberToWords(number);

  // Print the result
  print("Output: $result");
}

String convertNumberToWords(int number) {
  List<String> words = [];

  // Convert each digit to words
  while (number > 0) {
    int digit = number % 10;
    words.add(getWordForDigit(digit));
    number ~/= 10;
  }

  // Reverse the list to get the correct order
  words = words.reversed.toList();

  // Join the words with space and return
  return words.join(' ');
}

String getWordForDigit(int digit) {
  switch (digit) {
    case 0:
      return 'Zero';
    case 1:
      return 'One';
    case 2:
      return 'Two';
    case 3:
      return 'Three';
    case 4:
      return 'Four';
    case 5:
      return 'Five';
    case 6:
      return 'Six';
    case 7:
      return 'Seven';
```

```
    case 8:
      return 'Eight';
    case 9:
      return 'Nine';
    default:
      return '';
  }
}
```

Output:

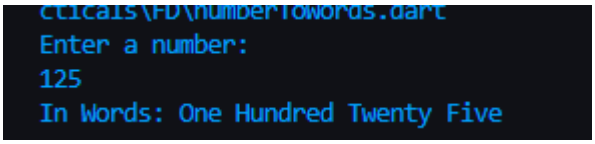


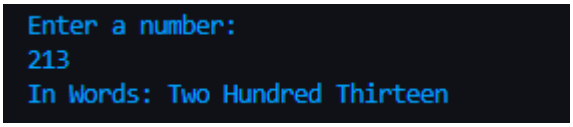
Q12. Write a program to print a number into words.

```
import 'dart:io';

void main() {
  print("Enter a number:");
  int number = int.parse(stdin.readLineSync() ?? '');

  String words = numberToWords(number);

  print("In Words: $words");
}

String numberToWords(int number) {
  if (number == 0) {
    return "Zero";
  }

  List<String> units = [
    "",
    "One",
    "Two",
    "Three",
    "Four",
    "Five",
    "Six",
    "Seven",
    "Eight",
    "Nine",
    "Ten",
    "Eleven",
    "Twelve",
    "Thirteen",
    "Fourteen",
    "Fifteen",
    "Sixteen",
    "Seventeen",
    "Eighteen",
    "Nineteen"
  ];

  List<String> tens = [
    "",
    "",
    "Twenty",
    "Thirty",
    "Forty",
    "Fifty",
    "Sixty",
    "Seventy",
    "Eighty",
```

```
    "Ninety"
  ];

  String result = "";

  if (number >= 100) {
    result += units[number ~/ 100] + " Hundred ";
    number %= 100;
  }

  if (number >= 20) {
    result += tens[number ~/ 10] + " ";
    number %= 10;
  }

  if (number > 0) {
    result += units[number];
  }

  return result.trim();
}
```

Output:

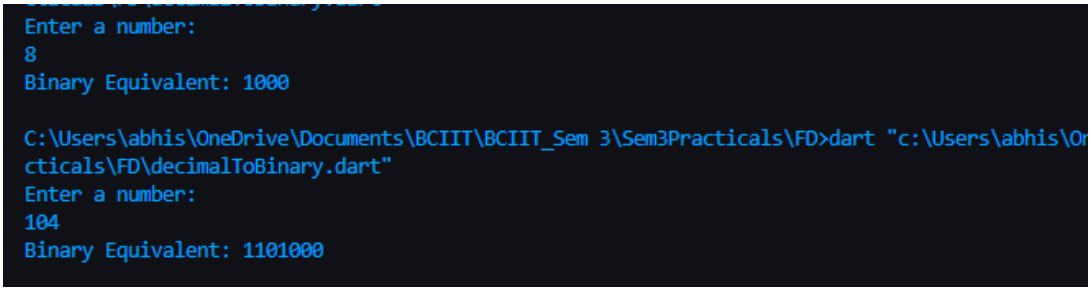




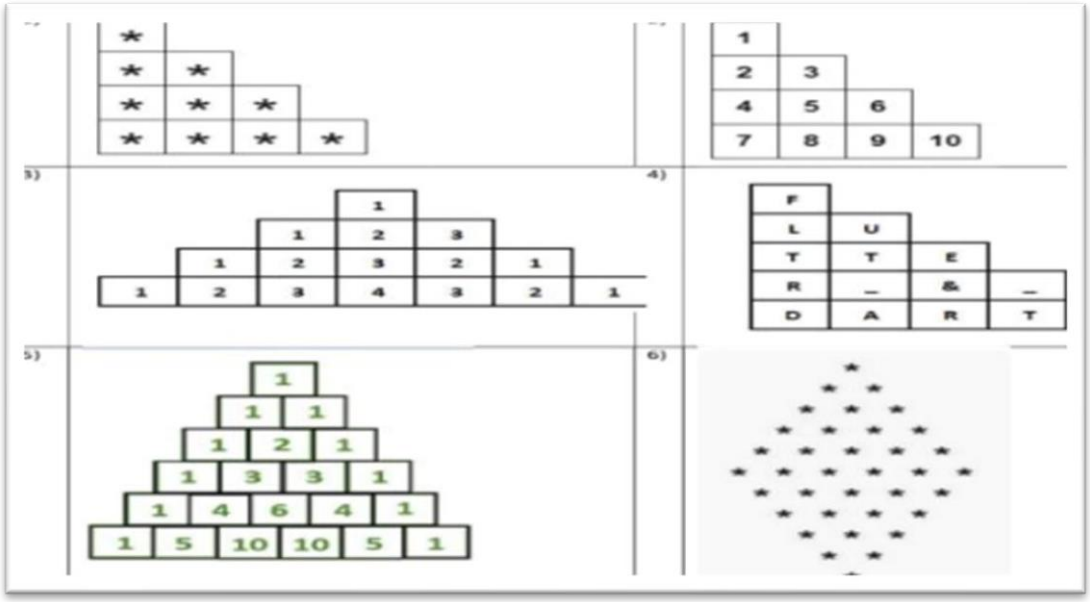
Q13. Write a program to print binary equivalent of a number.

```
import 'dart:io';

void main() {
  print("Enter a number:");
  int number = int.parse(stdin.readLineSync() ?? '');

  String binaryEquivalent = decimalToBinary(number);

  print("Binary Equivalent: $binaryEquivalent");
}

String decimalToBinary(int number) {
  if (number == 0) {
    return "0";
  }

  String binary = "";
  while (number > 0) {
    int remainder = number % 2;
    binary = "$remainder$binary";
    number ~/= 2;
  }

  return binary;
}
```

Output:



Q14. Write a program to print the given patterns.



Code-1:

```
import 'dart:io';

void main() {
  stdout.write("Enter the number of rows: ");
  int numRows = int.parse(stdin.readLineSync() ?? '');

  for (int i = 1; i <= numRows; i++) {
    for (int j = 1; j <= i; j++) {
    stdout.write("* ");
    }
    stdout.write("\n");
  }
}
```

Output:

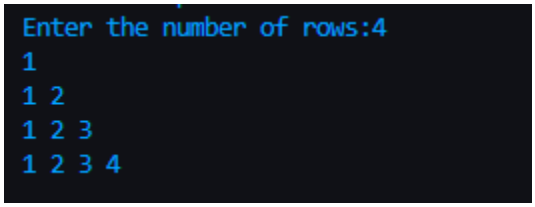
```
Enter the number of rows: 5
*
* *
* * *
* * * *
* * * * *
```

Code-2:

```
import 'dart:io';

void main() {
  stdout.write("Enter the number of rows:");
  int numRows = int.parse(stdin.readLineSync() ?? '');
```

```
  for (int i = 1; i <= numRows; i++) {
    for (int j = 1; j <= i; j++) {
    stdout.write("$j ");
    }
    stdout.write("\n");
  }
}
```

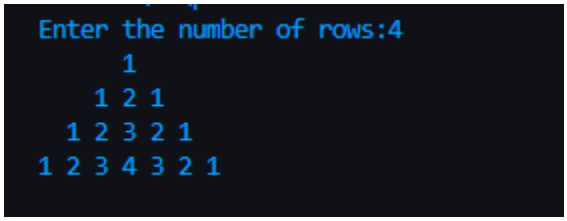
Output:



Code-3:

```
import 'dart:io';

void main() {
  stdout.write("Enter the number of rows:");
  int numRows = int.parse(stdin.readLineSync() ?? '');

  for (int i = 1; i <= numRows; i++) {
    for (int j = 1; j <= numRows - i; j++) {
      stdout.write("  ");
    }
    for (int j = 1; j <= i; j++) {
      stdout.write("$j ");
    }
    for (int j = i - 1; j >= 1; j--) {
      stdout.write("$j ");
    }
    stdout.write("\n");
  }
}
```

Output:

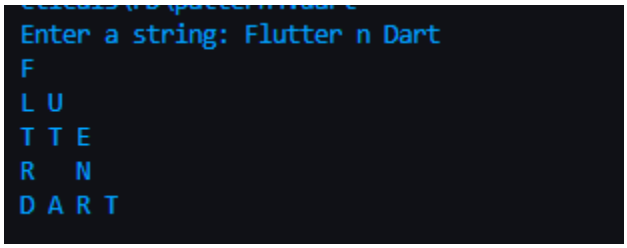


Code-4:

```
import 'dart:io';
```

```
void main() {
  stdout.write("Enter a string: ");
  String input = stdin.readLineSync() ?? '';
  input = input.toUpperCase();

  int length = input.length;
  int currentIndex = 0;

  for (int i = 1; currentIndex < length; i++) {
    for (int j = 0; j < i && currentIndex < length; j++) {
      stdout.write(input[currentIndex] + ' ');
      currentIndex++;
    }
    stdout.write('\n');
  }
}
```

Output:

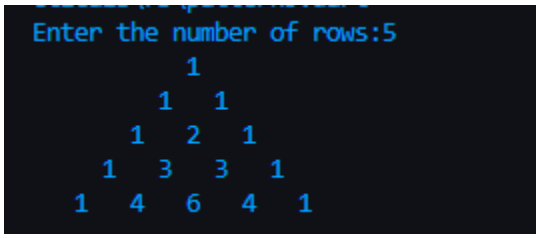


Code-5:

```
import 'dart:io';

void main() {
  stdout.write("Enter the number of rows:");
  int rows = int.parse(stdin.readLineSync() ?? '');

  for (int i = 0; i < rows; i++) {
    for (int space = 1; space < rows - i; ++space) {
      stdout.write("  ");
    }

    int coef = 1;
    for (int j = 0; j <= i; j++) {
      if (j == 0 || i == 0) {
        coef = 1;
      } else {
        coef = coef * (i - j + 1) ~/ j;
      }

      stdout.write("${coef.toString().padLeft(4)}");
    }

    stdout.write("\n");
```

```
  }
}
```

Output:

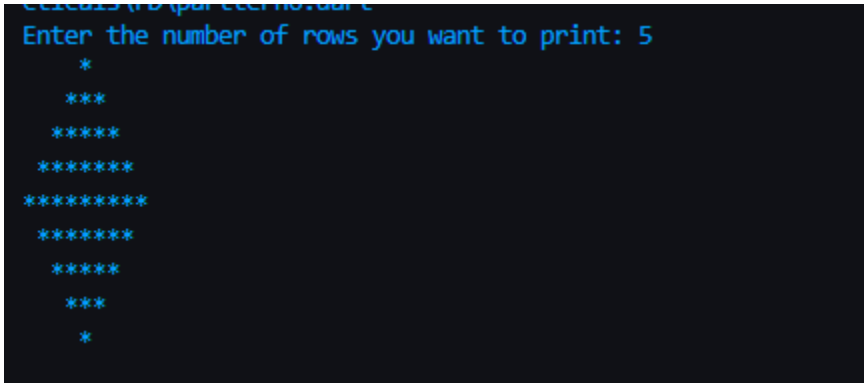


Code-6:

```
import 'dart:io';

void main() {
  stdout.write("Enter the number of rows you want to print: ");
  int row = int.parse(stdin.readLineSync() ?? '');

  if (row % 2 == 0) {
    row++; // Ensure odd number of rows
  }

  int space = row - 1;

  for (int j = 1; j <= row; j++) {
    for (int i = 1; i <= space; i++) {
    stdout.write(" ");
    }
    space--;

    for (int i = 1; i <= 2 * j - 1; i++) {
      stdout.write("*");
    }

    stdout.write("\n");
  }

  space = 1;
  for (int j = 1; j <= row - 1; j++) {
    for (int i = 1; i <= space; i++) {
    stdout.write(" ");
    }
    space++;

    for (int i = 1; i <= 2 * (row - j) - 1; i++) {
      stdout.write("*");
    }

    stdout.write("\n");
```

```
  }
}
```

Output:



Q15. Write a program to create the Application to perform the following operations on the number: Increment, Decrement, Square, Reset Check Prime, Check Even/Odd, Check Palindrome, Compute Factorial.

```
import 'package:flutter/material.dart';
import 'dart:math';

void main() {
  runApp(FirstFlutterApp());
}

class FirstFlutterApp extends StatefulWidget {
  @override
  _FirstFlutterAppState createState() => _FirstFlutterAppState();
}

class _FirstFlutterAppState extends State<FirstFlutterApp> {
  int incrementCount = 0;
  int decrementCount = 0;
  int squareCount = 0;
  int primeCount = 0;
  int evenOddCount = 0;
  int palindromeCount = 0;
  int factorialCount = 0;
  int userInput = 0;

  String resultText = '';

  bool isPrime(int number) {
    if (number <= 1) return false;
    if (number == 2) return true;
    for (int i = 2; i <= sqrt(number); i++) {
      if (number % i == 0) return false;
    }
    return true;
  }

  bool isEven(int number) {
    return number % 2 == 0;
  }

  bool isPalindrome(int number) {
    String numberStr = number.toString();
    String reversedNumberStr = numberStr.split('').reversed.join('');
    return numberStr == reversedNumberStr;
  }

  int calculateFactorial(int number) {
    if (number == 0 || number == 1) return 1;
    return number * calculateFactorial(number - 1);
  }
```

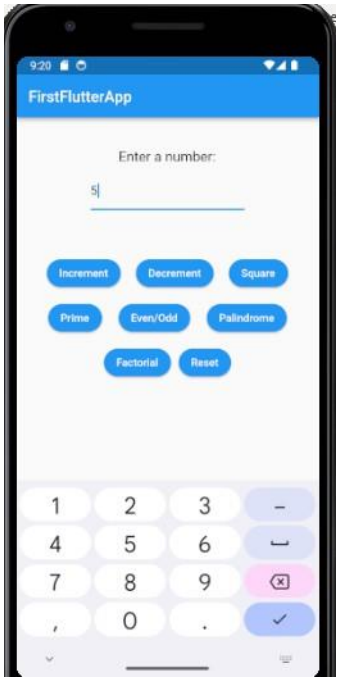
```
int findSquare(int number){
  return number*number;
}

void updateResultText(String action, int value) {
  setState(() {
    resultText =
    "You have pushed the $action button this many times: $value\n";
  });
}

void onButtonPress(String action) {
  setState(() {
    switch (action) {
      case 'Increment':
        incrementCount++;
        userInput++;
        updateResultText('increment', incrementCount);
        resultText +='Incremented to $userInput\n';
        break;
      case 'Decrement':
        decrementCount++;
        userInput--;
        updateResultText('decrement', decrementCount);
        resultText +='Decremented to $userInput\n';
        break;
      case 'Square':
        squareCount++;
        updateResultText('square', squareCount);
        resultText +='Square of $userInput is ${findSquare(userInput)}\n';
        break;
      case 'Prime':
        primeCount++;
        updateResultText('prime', primeCount);
        resultText +=
        '$userInput is ${isPrime(userInput) ? 'prime' : 'not prime'}\n';
        break;
      case 'Even/Odd':
        evenOddCount++;
        updateResultText('even/odd', evenOddCount);
        resultText +=
        '$userInput is ${isEven(userInput) ? 'even' : 'odd'}\n';
        break;
      case 'Palindrome':
        palindromeCount++;
        updateResultText('palindrome', palindromeCount);
        resultText +=
        '$userInput is ${isPalindrome(userInput) ? 'palindrome' : 'not palindrome'}\n';
        break;
      case 'Factorial':
```

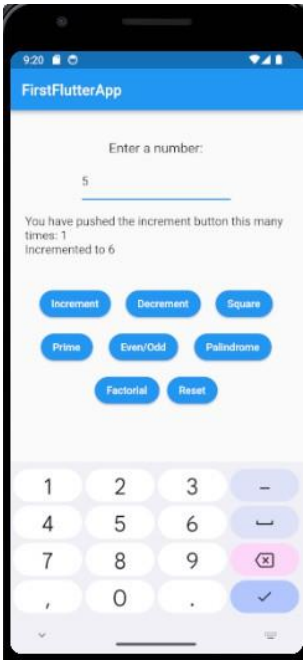
```
        factorialCount++;
        updateResultText('factorial', factorialCount);
        resultText +=
        'Factorial of $userInput is ${calculateFactorial(userInput)}\n';
        break;
      case 'Reset':
        incrementCount = 0;
        decrementCount = 0;
        squareCount = 0;
        primeCount = 0;
        evenOddCount = 0;
        palindromeCount = 0;
        factorialCount = 0;
        resultText = '';
        // Reset userInput to its original value
        userInput =0;
        break;
    }
  });
}

@override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'FirstFlutterApp',
    debugShowCheckedModeBanner: false,
    home: Scaffold(
      appBar: AppBar(
        title: Text('FirstFlutterApp'),
      ),
      body: SingleChildScrollView(
        padding: EdgeInsets.all(20),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: <Widget>[
            SizedBox(height: 20),
            Text(
              'Enter a number:',
              style: TextStyle(fontSize: 18),
            ),
            SizedBox(height: 10),
            Container(
              width: 200, // Set a specific width
              child: TextField(
                keyboardType: TextInputType.number,
                onChanged: (value) {
                  userInput = int.tryParse(value) ?? 0;
                },
              ),
            ),
            SizedBox(height: 20),
```

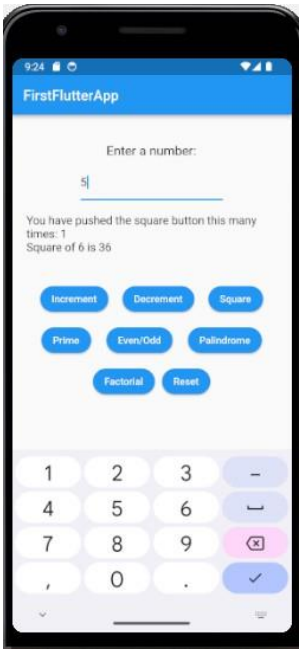
```
            Text(
              resultText,
              style: TextStyle(fontSize: 16),
            ),
            SizedBox(height: 20),
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: [
                buildButton('Increment'),
                buildButton('Decrement'),
                buildButton('Square'),
              ],
            ),
            SizedBox(height: 10),
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: [
                buildButton('Prime'),
                buildButton('Even/Odd'),
                buildButton('Palindrome'),
              ],
            ),
            SizedBox(height: 10),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                buildButton('Factorial'),
                SizedBox(width: 10),
                buildButton('Reset'),
              ],
            ),
          ],
        ),
      ),
    ),
  );
}

Widget buildButton(String action) {
  return ElevatedButton(
    onPressed: () {
    onButtonPress(action);
    },
    style: ButtonStyle(
      backgroundColor: MaterialStateProperty.all<Color>(Colors.blue),
      shape: MaterialStateProperty.all<OutlinedBorder>(
      RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(30.0),
        ),
      ),
    ),
```

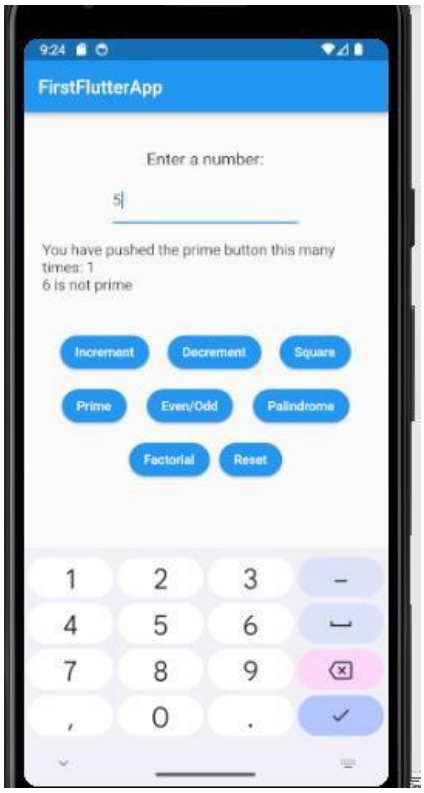
```
      child: Text(action),
    );
  }
}
```

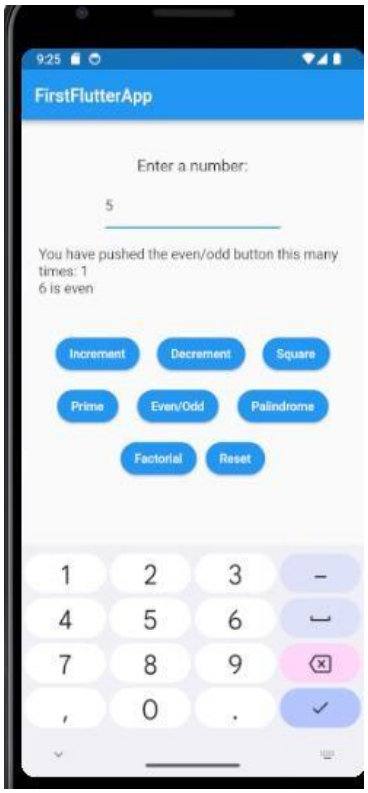
Output:

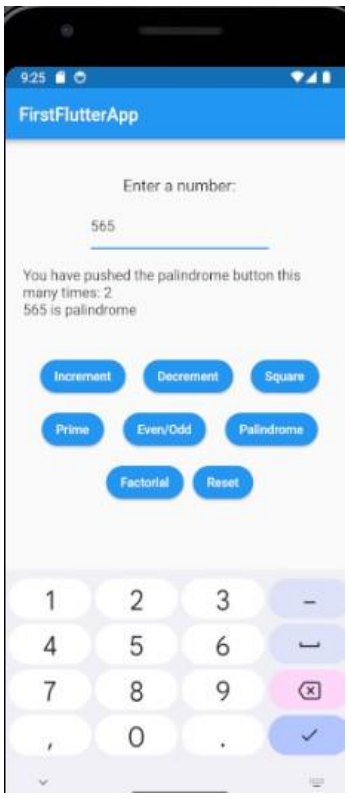


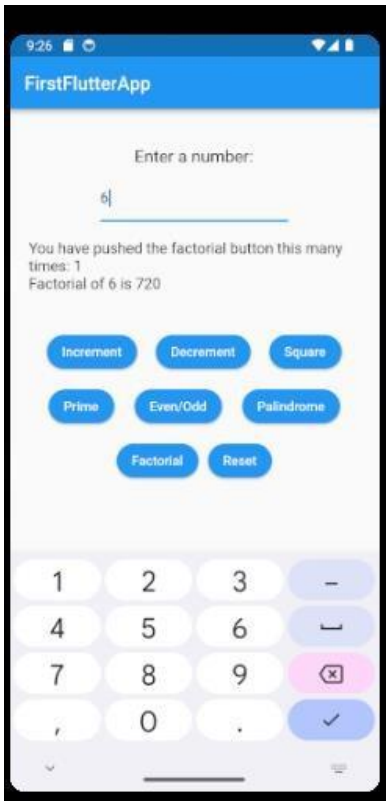


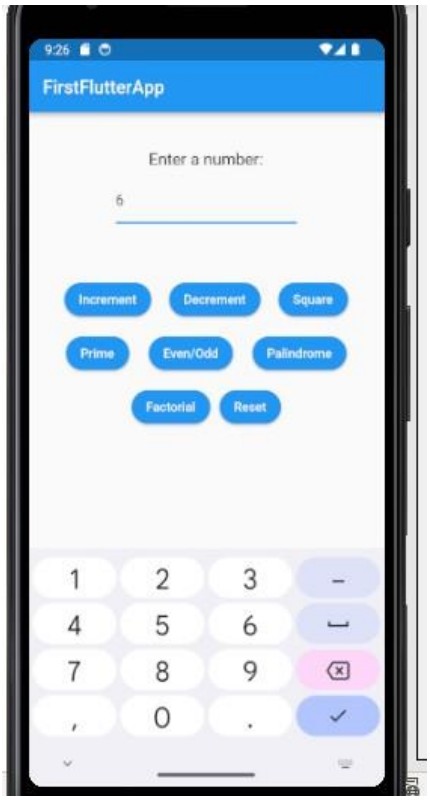









9:26
FirstFlutterApp
Enter a number:
6
You have pushed the factorial button this many times: 1
Factorial of 6 is 720
Increment
Decrement
Square
Prime
Even/Odd
Palindrome
Factorial
Reset


9:26
FirstFlutterApp
Enter a number:
6
Increment
Decrement
Square
Prime
Even/Odd
Palindrome
Factorial
Reset

Q16. Create a flutter application using Bottom Navigation Bar, Bottom app Bar and Tab Bar.

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> with SingleTickerProviderStateMixin {
  late TabController _tabController;

  @override
  void initState() {
    super.initState();
    _tabController = TabController(length: 4, vsync: this);
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Bottom Navigation Bar Demo'),
      ),
      body: DefaultTabController(
        length: 4,
        child: Column(
          children: [
          Container(
              color: Colors.blue,
              child: TabBar(
                controller: _tabController,
                labelColor: Colors.white,
                unselectedLabelColor: Colors.white.withOpacity(0.5),
                tabs: [
                  Tab(icon: Icon(Icons.call), text: 'Incoming'),
                  Tab(icon: Icon(Icons.call_made), text: 'Outgoing'),
```

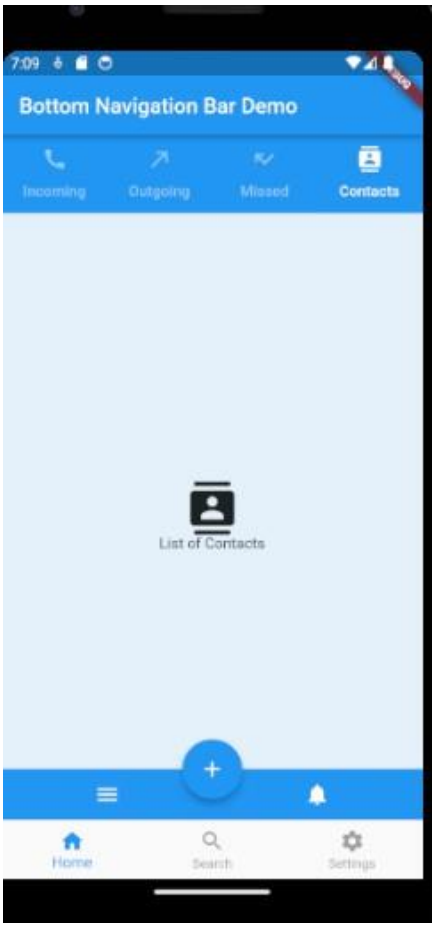
```
      Tab(icon: Icon(Icons.call_missed), text: 'Missed'),
      Tab(icon: Icon(Icons.contacts), text: 'Contacts'),
    ],
  ),
),
Expanded(
  child: Container(
    color: Colors.blue.withOpacity(0.1),
    child: TabBarView(
      controller: _tabController,
      children: [
        // Incoming Tab
        Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Icon(Icons.call_received, size: 50),
              Text('List of Incoming Calls'),
            ],
          ),
        ),
        // Outgoing Tab
        Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Icon(Icons.call_made, size: 50),
              Text('List of Outgoing Calls'),
            ],
          ),
        ),
        // Missed Tab
        Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Icon(Icons.call_missed, size: 50),
              Text('List of Missed Calls'),
            ],
          ),
        ),
        // Contacts Tab
        Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Icon(Icons.contacts, size: 50),
              Text('List of Contacts'),
            ],
          ),
        ),
```

```
            ],
          ),
        ),
      ),
    ],
  ),
),
bottomNavigationBar: BottomNavigationBar(
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.search),
      label: 'Search',
    ),
    BottomNavigationBarItem(
      icon:  Icon(Icons.settings),
      label: 'Settings',
    ),
  ],
  selectedItemColor: Colors.blue,
  unselectedItemColor: Colors.grey,
  currentIndex: 0,
  onTap: (index) {},
),
floatingActionButton: FloatingActionButton(
  onPressed: () {},
  child: Icon(Icons.add),
  backgroundColor: Colors.blue,
),
floatingActionButtonLocation: FloatingActionButtonLocation.centerDocked,
bottomSheet: BottomAppBar(
  color: Colors.blue,
  shape: CircularNotchedRectangle(),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceAround,
    children: [
      IconButton(
        icon: Icon(Icons.menu),
        color: Colors.white,
        onPressed: () {},
      ),
      IconButton(
        icon: Icon(Icons.notifications),
        color: Colors.white,
        onPressed: () {},
      ),
    ],
  ),
```

```
      ),
    );
  }
}
```

Output:


7:08
Bottom Navigation Bar Demo
Incoming
Outgoing
Missed
Contacts
List of Incoming Calls
Home
Search
Settings


7:09
Bottom Navigation Bar Demo
Incoming
Outgoing
Missed
Contacts
List of Contacts
Home
Search
Settings

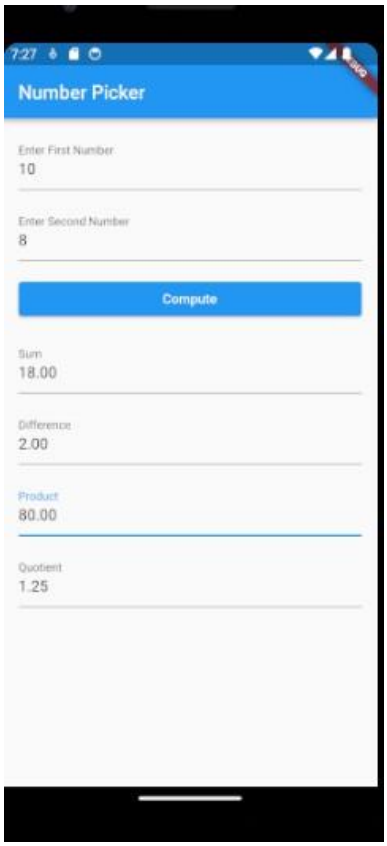
Q17. Create a flutter application that obtains two integers from the user and prints their product, difference, and quotient (division).

```
import 'package:flutter/material.dart';

void main() {
  runApp(NumberPickerApp());
}

class NumberPickerApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: NumberPickerScreen(),
    );
  }
}

class NumberPickerScreen extends StatefulWidget {
  @override
  _NumberPickerScreenState createState() => _NumberPickerScreenState();
}

class _NumberPickerScreenState extends State<NumberPickerScreen> {
  TextEditingController firstNumberController = TextEditingController();
  TextEditingController secondNumberController = TextEditingController();
  TextEditingController sumController = TextEditingController();
  TextEditingController differenceController = TextEditingController();
  TextEditingController productController = TextEditingController();
  TextEditingController quotientController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Number Picker'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            TextField(
              controller: firstNumberController,
              keyboardType: TextInputType.number,
              decoration: InputDecoration(labelText: 'Enter First Number'),
            ),
            SizedBox(height: 16),
            TextField(
              controller: secondNumberController,
              keyboardType: TextInputType.number,
```

```
            decoration: InputDecoration(labelText: 'Enter Second Number'),
          ),
          SizedBox(height: 16),
          ElevatedButton(
          onPressed: () {
             calculateValues();
            },
            child: Text('Compute'),
          ),
          SizedBox(height: 16),
          TextField(
            controller: sumController,
            readOnly: true,
            decoration: InputDecoration(labelText: 'Sum'),
          ),
          SizedBox(height: 16),
          TextField(
            controller: differenceController,
            readOnly: true,
            decoration: InputDecoration(labelText: 'Difference'),
          ),
          SizedBox(height: 16),
          TextField(
            controller: productController,
            readOnly: true,
            decoration: InputDecoration(labelText: 'Product'),
          ),
          SizedBox(height: 16),
          TextField(
            controller: quotientController,
            readOnly: true,
            decoration: InputDecoration(labelText: 'Quotient'),
          ),
        ],
      ),
    ),
  );
}

void calculateValues() {
  double firstNumber = double.tryParse(firstNumberController.text) ?? 0.0;
  double secondNumber = double.tryParse(secondNumberController.text) ?? 0.0;

  double sum = firstNumber + secondNumber;
  double difference = firstNumber - secondNumber;
  double product = firstNumber * secondNumber;

  // Avoid division by zero
  double quotient = secondNumber != 0.0 ? firstNumber / secondNumber : 0.0;

  sumController.text = sum.toStringAsFixed(2);
```

```
    differenceController.text = difference.toStringAsFixed(2);
    productController.text = product.toStringAsFixed(2);
    quotientController.text = quotient.toStringAsFixed(2);
  }
}
```

Output:



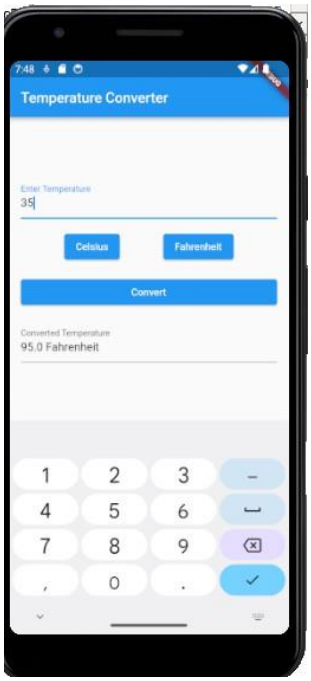
Q18. Create a flutter application that converts a number from Fahrenheit to degree and vice-versa.

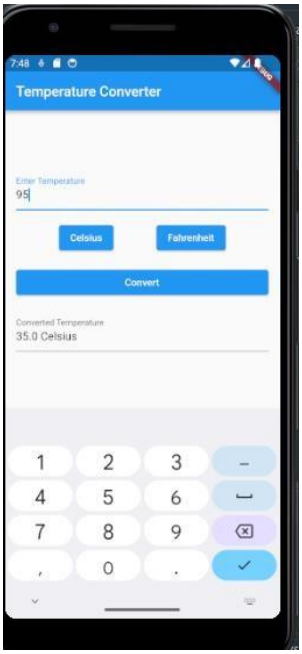
```
import 'package:flutter/material.dart';

void main() {
  runApp(TemperatureConverterApp());
}

class TemperatureConverterApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: TemperatureConverterScreen(),
    );
  }
}

class TemperatureConverterScreen extends StatefulWidget {
  @override
  _TemperatureConverterScreenState createState() =>
      _TemperatureConverterScreenState();
}

class _TemperatureConverterScreenState
    extends State<TemperatureConverterScreen> {
  TextEditingController userInputController = TextEditingController();
  TextEditingController resultController = TextEditingController();
  String selectedUnit = 'Celsius';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Temperature Converter'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            TextField(
              controller: userInputController,
              keyboardType: TextInputType.number,
              decoration: InputDecoration(
              labelText: 'Enter Temperature',
              ),
            ),
            SizedBox(height: 16),
            Row(
```

```
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              ElevatedButton(
                onPressed: () {
                selectUnit('Celsius');
                },
                child: Text('Celsius'),
              ),
              ElevatedButton(
                onPressed: () {
                selectUnit('Fahrenheit');
                },
                child: Text('Fahrenheit'),
              ),
            ],
          ),
          SizedBox(height: 16),
          ElevatedButton(
          onPressed: () {
              compute();
            },
            child: Text('Convert'),
          ),
          SizedBox(height: 16),
          TextField(
            controller: resultController,
            readOnly: true,
            decoration: InputDecoration(
              labelText: 'Converted Temperature',
            ),
          ),
        ],
      ),
    ),
  );
}

void selectUnit(String unit) {
  setState(() {
    selectedUnit = unit;
  });
}

void compute() {
  double inputTemperature =
      double.tryParse(userInputController.text) ?? 0.0;

  double convertedTemperature;
  String unit;

  if (selectedUnit == 'Celsius') {
```

```
      convertedTemperature = (inputTemperature - 32) * 5 / 9;
      unit = 'Celsius';
    } else {
      convertedTemperature = (inputTemperature * 9 / 5) + 32;
      unit = 'Fahrenheit';
    }

    setState(() {
      resultController.text = '$convertedTemperature $unit';
    });
  }
}
```

Output:





Q19. Create a flutter application with UI to display the image of the image URL in TextField.

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: Home(),
    );
  }
}

class Home extends StatefulWidget {
  const Home({Key? key}) : super(key: key);

  @override
  _HomeState createState() => _HomeState();
}
class _HomeState extends State<Home> {
  final titleController = TextEditingController();
  String imgUrl = "";

  void _setImage() {
    setState(() {
      imgUrl = titleController.text;
    });
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Image Example"),
        backgroundColor: Colors.pinkAccent,
      ),
      body: Column(
        children: [
        Padding(
            padding: const EdgeInsets.all(15),
            child: TextField(
              decoration: const InputDecoration(labelText: 'Image URL'),
              controller: titleController,
            ),
          ),
```

```
          const SizedBox(
            height: 8,
          ),
          ElevatedButton(
            onPressed: _setImage,
            style: ButtonStyle(
              elevation: MaterialStateProperty.all(8),
              backgroundColor: MaterialStateProperty.all(Colors.pinkAccent),
            ),
            child: const Text('Set Image'),
          ),
          const SizedBox(
            height: 20,
          ),
          imgUrl.isNotEmpty
              ? Image.network(
            imgUrl,
            height: 200,
            width: 200,
          )
              : const Text('Enter an image URL and press "Set Image" to display.'),
        ],
      ),
    );
  }
}
```

Output:

Q20. Create a flutter application using Inkwell , OnTap event widgets.

```
import 'dart:math';

import 'package:flutter/material.dart';

void main() {
  runApp(DiceGameApp());
}

class DiceGameApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: DiceGameScreen(),
    );
  }
}

class DiceGameScreen extends StatefulWidget {
  @override
  _DiceGameScreenState createState() => _DiceGameScreenState();
}

class _DiceGameScreenState extends State<DiceGameScreen> {
  int player1Score = 0;
  int player2Score = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Dice Game'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'Player 1: $player1Score',
              style: TextStyle(fontSize: 20),
            ),
            SizedBox(height: 20),
            InkWell(
              onTap: () {
                rollDice(1);
              },
              child:  Image.asset(
                'assets/dice.PNG',
                height: 150,
                width: 150,
```

```
            ),
          ),
          SizedBox(height: 50),
          Text(
            'Player 2: $player2Score',
            style: TextStyle(fontSize: 20),
          ),
          SizedBox(height: 20),
          InkWell(
            onTap: () {
              rollDice(2);
            },
            child:  Image.asset(
              'assets/dice.PNG',
              height: 150,
              width: 150,
            ),
          ),
          SizedBox(height: 50),
          ElevatedButton(
          onPressed: () {
              resetGame();
            },
            child: Text('Reset Game'),
          ),
        ],
      ),
    ),
  );
}

void rollDice(int player) {
  setState(() {
    int diceValue = Random().nextInt(6) + 1;
    if (player == 1) {
      player1Score = diceValue;
    } else {
      player2Score = diceValue;
    }
    checkWinner();
  });
}

void checkWinner() {
  if (player1Score > 0 && player2Score > 0) {
    String winner = player1Score > player2Score ? 'Player 1' : 'Player 2';
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: Text('Winner'),
        content: Text('$winner wins!'),
```

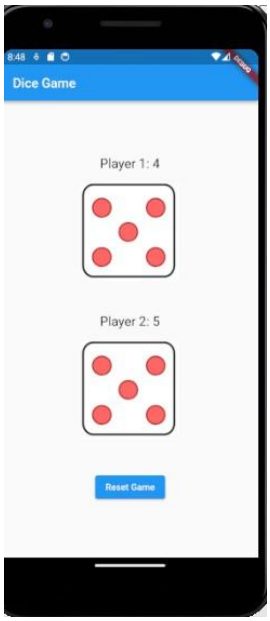
```
          actions: [
            TextButton(
            onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text('OK'),
            ),
          ],
        ),
      );
    }
  }

  void resetGame() {
    setState(() {
    player1Score = 0;
      player2Score = 0;
    });
  }
}
```

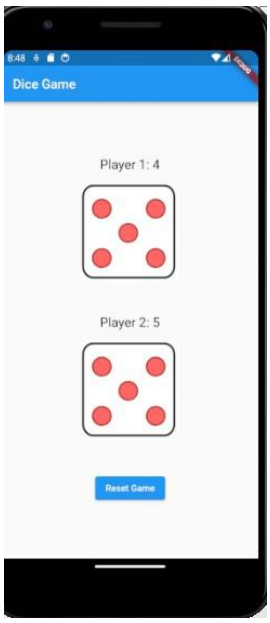
pubspec.yaml:

```
name: flutterpracs
description: A new Flutter project.

publish_to: 'none' # Remove this line if you wish to publish to pub.dev


version: 1.0.0+1

environment:
  sdk: '>=3.1.3 <4.0.0'

dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.2

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^2.0.0


flutter:


  uses-material-design: true
```

```
assets:
  - app_img_src/
  - assets/
```

Output:





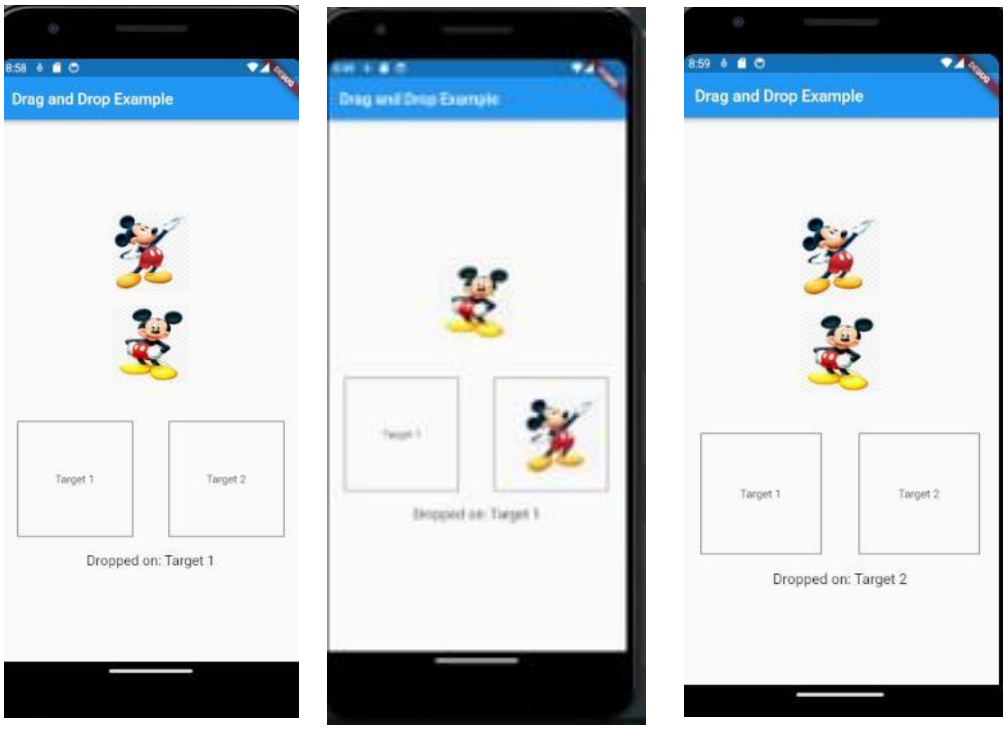
Q21. Create a flutter application using Draggable, Dragtarget widgets.

```
import 'package:flutter/material.dart';

void main() {
  runApp(DragDropApp());
}

class DragDropApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: DragDropScreen(),
    );
  }
}

class DragDropScreen extends StatefulWidget {
  @override
  _DragDropScreenState createState() => _DragDropScreenState();
}

class _DragDropScreenState extends State<DragDropScreen> {
  String droppedOnTarget = '';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Drag and Drop Example'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Draggable<String>(
              data: 'mickey1',
              child: Image.asset(
                'assets/mickey1.png',
                height: 100,
                width: 100,
              ),
              feedback: Image.asset(
                'assets/mickey1.png',
                height: 100,
                width: 100,
              ),
              childWhenDragging: Container(),
            ),
            SizedBox(height: 20),
            Draggable<String>(
```

```
            data: 'mickey2',
            child: Image.asset(
              'assets/mickey2.png',
              height: 100,
              width: 100,
            ),
            feedback: Image.asset(
              'assets/mickey2.png',
              height: 100,
              width: 100,
            ),
            childWhenDragging: Container(),
          ),
          SizedBox(height: 50),
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: [
              buildDragTarget('Target 1'),
              buildDragTarget('Target 2'),
            ],
          ),
          SizedBox(height: 20),
          Text(
            'Dropped on: $droppedOnTarget',
            style: TextStyle(fontSize: 18),
          ),
        ],
      ),
    ),
  );
}

Widget buildDragTarget(String target) {
  return DragTarget<String>(
    builder: (BuildContext context, List<String?> candidateData, List<dynamic> rejectedData) {
      return Container(
        width: 150,
        height: 150,
        decoration: BoxDecoration(
          border: Border.all(color: Colors.black),
        ),
        child: Center(
          child: Text(target),
        ),
      );
    },
    onWillAccept: (data) {
      return true;
    },
    onAccept: (data) {
      setState(() {
```

```
          droppedOnTarget = target;
        });
      },
      onLeave: (data) {},
    );
  }
}
```

Output:



Q22. Create a Tic Tac Toe Application.

```
import 'package:flutter/material.dart';

void main() {
  runApp(TicTacToeApp());
}

class TicTacToeApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: TicTacToeScreen(),
    );
  }
}

class TicTacToeScreen extends StatefulWidget {
  @override
  _TicTacToeScreenState createState() => _TicTacToeScreenState();
}

class _TicTacToeScreenState extends State<TicTacToeScreen> {
  List<List<String>> board = List.generate(3, (_) => List.filled(3, ''));

  bool player1Turn = true;
  int player1Score = 0;
  int player2Score = 0;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Tic Tac Toe'),
      ),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          buildScoreBoard(),
          buildGameBoard(),
          buildResetButton(),
        ],
      ),
    );
  }

  Widget buildScoreBoard() {
    return Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text('Player 1: $player1Score'),
```

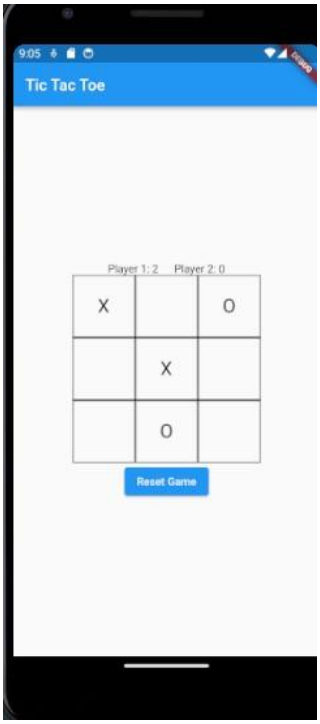
```
      SizedBox(width: 20),
      Text('Player 2: $player2Score'),
    ],
  );
}

Widget buildGameBoard() {
  return Column(
    children: List.generate(3, (row) {
      return Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: List.generate(3, (col) {
          return GestureDetector(
            onTap: () {
              if (board[row][col].isEmpty) {
                setState(() {
                  board[row][col] = player1Turn ? 'X' : 'O';
                  checkWinner();
                  player1Turn = !player1Turn;
                });
              }
            },
            child: Container(
              width: 80,
              height: 80,
              decoration: BoxDecoration(
                border: Border.all(color: Colors.black),
              ),
              child: Center(
                child: Text(
                board[row][col],
                  style: TextStyle(fontSize: 24),
                ),
              ),
            ),
          );
        }),
      );
    }),
  );
}

Widget buildResetButton() {
  return ElevatedButton(
  onPressed: () {
      setState(() {
        resetGame();
      });
    },
    child: Text('Reset Game'),
  );
```

```
}

void checkWinner() {
  // Check rows
  for (int i = 0; i < 3; i++) {
    if (board[i][0] == board[i][1] && board[i][1] == board[i][2] && board[i][0].isNotEmpty) {
      showWinnerDialog(board[i][0]);
      return;
    }
  }

  // Check columns
  for (int i = 0; i < 3; i++) {
    if (board[0][i] == board[1][i] && board[1][i] == board[2][i] && board[0][i].isNotEmpty) {
      showWinnerDialog(board[0][i]);
      return;
    }
  }

  // Check diagonals
  if (board[0][0] == board[1][1] && board[1][1] == board[2][2] && board[0][0].isNotEmpty) {
    showWinnerDialog(board[0][0]);
    return;
  }

  if (board[0][2] == board[1][1] && board[1][1] == board[2][0] && board[0][2].isNotEmpty) {
    showWinnerDialog(board[0][2]);
    return;
  }

  // Check for a draw
  if (!board.any((row) => row.any((cell) => cell.isEmpty))) {
    showDrawDialog();
  }
}

void showWinnerDialog(String winner) {
  String message = winner == 'X' ? 'Player 1 wins!' : 'Player 2 wins!';
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: Text('Game Over'),
      content: Text(message),
      actions: [
        TextButton(
        onPressed: () {
            Navigator.of(context).pop();
            resetGame();
          },
          child: Text('OK'),
        ),
```

```
        ],
      ),
    );
    if (winner == 'X') {
      player1Score++;
    } else {
      player2Score++;
    }
  }

  void showDrawDialog() {
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        title: Text('Game Over'),
        content: Text('It\'s a draw!'),
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
              resetGame();
            },
            child: Text('OK'),
          ),
        ],
      ),
    );
  }

  void resetGame() {
    setState(() {
      board = List.generate(3, (_) => List.filled(3, ''));
      player1Turn = true;
    });
  }
}
```

‘

Output:

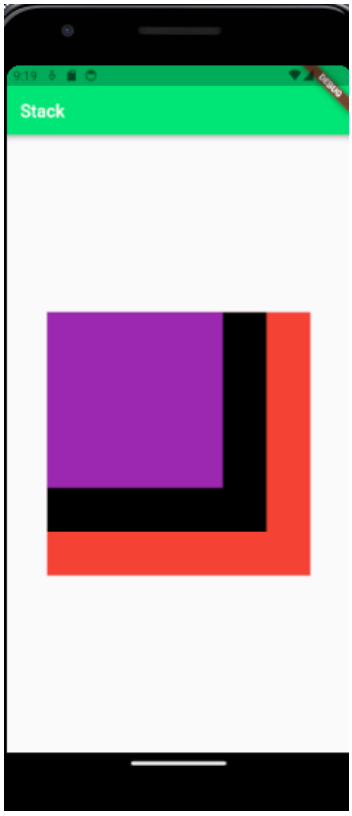

9:05
Tic Tac Toe
Player 1: 2 Player 2: 0
X
O
X
O
Reset Game


9:05
Tic Tac Toe
Player 1: 3 Player 2: 0
Game Over
Player 1 wins!
OK
Reset Game

Q23. Create a flutter application to using stack widget.

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
      home: Scaffold(
          appBar: AppBar(
          title: Text('Stack'),
            backgroundColor: Colors.greenAccent[400],
          ), //AppBar
          body: Center(
            child: SizedBox(
            width: 300,
              height: 300,
              child: Center(
              child: Stack(
                  children: <Widget>[
                    Container(
                      width: 300,
                      height: 300,
                      color: Colors.red,
                    ), //Container
                    Container(
                    width: 250,
                      height: 250,
                      color: Colors.black,
                    ), //Container
                    Container(
                    height: 200,
                      width: 200,
                      color: Colors.purple,
                    ), //Container
                  ], //<Widget>[]
                ), //Stack
              ), //Center
            ), //SizedBox
          ) //Center
      ) //Scaffold
  ) //MaterialApp
  );
}
```

Output



Q24. Design a login Page and navigate to the next page if correct credentials are entered.

main.dart:

```
//main.dart
import 'package:flutter/material.dart';
import 'homePage.dart'; // Import the HomePage widget
import 'valid.dart';
import 'invalid.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      initialRoute: '/home', // Set the initial route
      routes: {
        '/home': (context) => homePage(),
        '/valid': (context) => valid(),
        '/invalid': (context) => invalid(),
      },
    );
  }
}
```

homepage.dart:

```
//homePage.dart
import 'package:flutter/material.dart';

class homePage extends StatefulWidget {
  @override
  _homePageState createState() => _homePageState();
}

class _homePageState extends State<homePage> {
  final nmController = TextEditingController();
  final pwdController = TextEditingController();
  String _usrAuth = '';

  void _authenticateUser(String message) {
    if (nmController.text == 'a' && pwdController.text == 'b') {
      // Valid login
      Navigator.pushNamed(context, '/valid');
    } else {
      // Invalid login
```

```
    Navigator.pushNamed(context, '/invalid');
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Login Page'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Text(
            'Login',
            style: TextStyle(
              color: Colors.blue,
              fontWeight: FontWeight.w500,
              fontSize: 30,
            ),
          ),
          Container(
            padding: const EdgeInsets.all(10),
            child: TextField(
              controller: nmController,
              decoration: InputDecoration(
              border: OutlineInputBorder(),
              labelText: 'User Name',
              ),
            ),
          ),
          Container(
            padding: const EdgeInsets.all(10),
            child: TextField(
              controller: pwdController,
              decoration: InputDecoration(
              border: OutlineInputBorder(),
              labelText: 'Password',
              ),
              obscureText: true, // Use a password field
            ),
          ),
          Container(
            height: 50,
            padding: const EdgeInsets.fromLTRB(10, 0, 10, 0),
            child: ElevatedButton(
              child: Text('Login'),
              onPressed: () {
                _authenticateUser("Valid User");
              },
```

```
            ),
          ),
          Text(
            '$_usrAuth',
            style: TextStyle(
              color: Colors.blue,
              fontWeight: FontWeight.w500,
              fontSize: 20,
            ),
          ),
        ],
      ),
    ),
  );
 }
}
```

valid.dart:

```
//valid.dart
import 'package:flutter/material.dart';

class valid extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Valid Page'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Icon(
            Icons.check_circle,
            size: 100,
            color: Colors.green,
          ),
          SizedBox(height: 20),
          Text(
            'You are logged in as a valid user.',
            style: TextStyle(
              fontSize: 18,
              fontWeight: FontWeight.bold,
            ),
          ),
        ],
      ),
    ),
  );
```

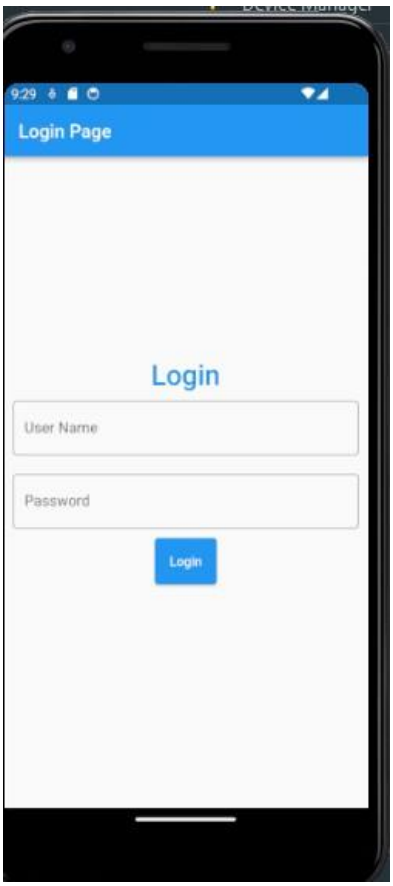
```
  }
}
```

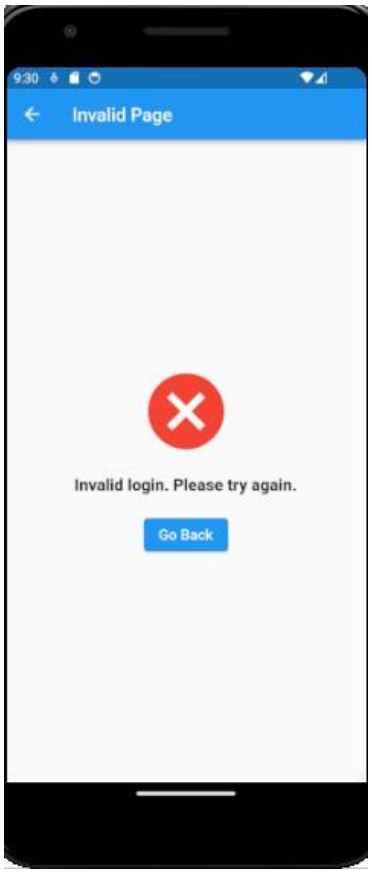
invalid.dart:

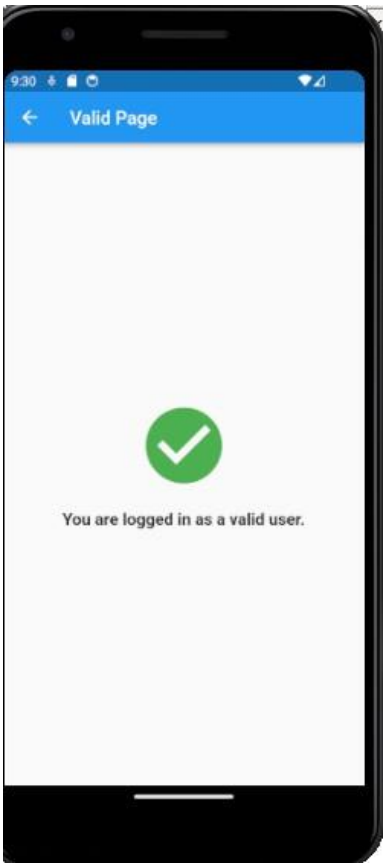
```
//invalid.dart
import 'package:flutter/material.dart';
import 'homePage.dart';

class invalid extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Invalid Page'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Icon(
              Icons.cancel,
              size: 100,
              color: Colors.red,
            ),
            SizedBox(height: 20),
            Text(
              'Invalid login. Please try again.',
              style: TextStyle(
                fontSize: 18,
                fontWeight: FontWeight.bold,
              ),
            ),
            SizedBox(height: 20),
            ElevatedButton(
            onPressed: () {
                // Navigate back to the HomePage
                Navigator.pushReplacementNamed(context,  '/home');
              },
              child: Text(
                'Go Back',
                style: TextStyle(
                  fontSize: 16,
                ),
              ),
            ),
          ],
        ),
      ),
    );
```

```
  }
}
```

Output:


9:29
Login Page
Login
User Name
Password
Login


9:30
Invalid Page
Invalid login. Please try again.
Go Back


9:30
Valid Page
You are logged in as a valid user.

Q25. Create a flutter application to play and pause a video.

```
import 'dart:async';

import 'package:flutter/material.dart';
import 'package:video_player/video_player.dart';

void main() => runApp(const VideoPlayerApp());

class VideoPlayerApp extends StatelessWidget {
  const VideoPlayerApp({super.key});

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      title: 'Video Player Demo',
      home: VideoPlayerScreen(),
    );
  }
}

class VideoPlayerScreen extends StatefulWidget {
  const VideoPlayerScreen({super.key});

  @override
  State<VideoPlayerScreen> createState() => _VideoPlayerScreenState();
}

class _VideoPlayerScreenState extends State<VideoPlayerScreen> {
  late VideoPlayerController _controller;
  late Future<void> _initializeVideoPlayerFuture;

  @override
  void initState() {
    super.initState();

    // Create and store the VideoPlayerController. The VideoPlayerController
    // offers several different constructors to play videos from assets, files,
    // or the internet.
    _controller = VideoPlayerController.networkUrl(
      Uri.parse(
        'https://flutter.github.io/assets-for-api-docs/assets/videos/butterfly.mp4',
      ),
    );

    // Initialize the controller and store the Future for later use.
    _initializeVideoPlayerFuture = _controller.initialize();

    // Use the controller to loop the video.
    _controller.setLooping(true);
  }
```

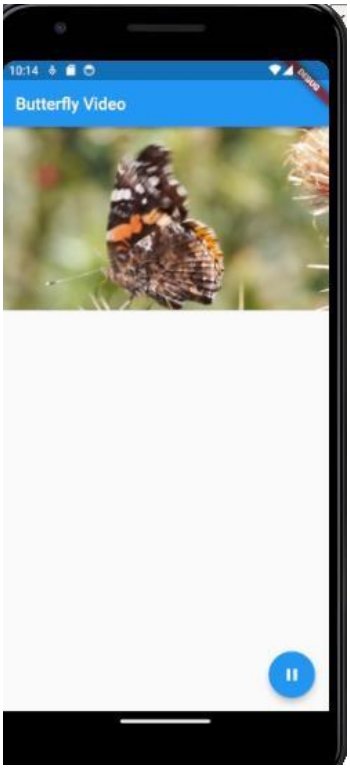
```
@override
void dispose() {
  // Ensure disposing of the VideoPlayerController to free up resources.
  _controller.dispose();

  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Butterfly Video'),
    ),
    // Use a FutureBuilder to display a loading spinner while waiting for the
    // VideoPlayerController to finish initializing.
    body: FutureBuilder(
      future: _initializeVideoPlayerFuture,
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.done) {
          // If the VideoPlayerController has finished initialization, use
          // the data it provides to limit the aspect ratio of the video.
          return AspectRatio(
            aspectRatio: _controller.value.aspectRatio,
            // Use the VideoPlayer widget to display the video.
            child: VideoPlayer(_controller),
          );
        } else {
          // If the VideoPlayerController is still initializing, show a
          // loading spinner.
          return const Center(
            child: CircularProgressIndicator(),
          );
        }
      },
    ),
    floatingActionButton: FloatingActionButton(
      onPressed: () {
        // Wrap the play or pause in a call to `setState`. This ensures the
        // correct icon is shown.
        setState(() {
          // If the video is playing, pause it.
          if (_controller.value.isPlaying) {
            _controller.pause();
          } else {
            // If the video is paused, play it.
            _controller.play();
          }
        });
      },
```

```
        // Display the correct icon depending on the state of the player.
        child: Icon(
          _controller.value.isPlaying ? Icons.pause : Icons.play_arrow,
        ),
      ),
    );
  }
}
```

Add the following permission to the AndroidManifest.xml file just after the <application> definition. The AndroidManifest.xml file is found at <project root>/android/app/src/main/AndroidManifest.xml.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <application ...>

    </application>

    <uses-permission android:name="android.permission.INTERNET"/>
</manifest>
```

Output:



Q26. Create the list of students and display using list view.

main.dart:

```
import 'dart:math';
import 'package:flutter/material.dart';
import 'header.dart';
import 'RowWidget.dart';
import 'RowWithCardWidget.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Mca 3rd - Card Demo',
      home: HomePage(),   ); }}

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}
class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('MCA 3rd sem List View Demo'),  ),
      body: SafeArea(
        child: ListView.builder(
          itemCount: 20,
          itemBuilder: (BuildContext context, int index){
            if(index == 0){
              return HeaderWidget(index: index);
            }
            else if (index >= 1 && index <= 3){
              return RowWithCardWidget(index: index);
            }
            else{
              return RowWidget(index: index);  }       },       ), ), ); }}
```

header.dart:

```
import 'package:flutter/material.dart';

class HeaderWidget extends StatelessWidget {
  const HeaderWidget({
    required this.index,
  });
```

```
  final int index;

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: EdgeInsets.all(16.0),
      height: 120.0,
      child: Card(
        elevation: 8.0,
        color: Colors.white,
        shape: RoundedRectangleBorder(), // Use RoundedRectangleBorder instead of StarBorder
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'R. Ponting',
              textAlign: TextAlign.center,
              style: TextStyle(
                fontWeight: FontWeight.bold,
                fontSize: 48.0,
                color: Colors.orange,
              ),
            ),
            Text(
              '04011104422',
              textAlign: TextAlign.center,
              style: TextStyle(
                color: Colors.grey,
              ),
            )
          ],
        ),
      ),
    );
  }
}
```

RowWidget.dart:

```
import 'package:flutter/material.dart';


class RowWidget extends StatelessWidget{
  const RowWidget({
    Key? key, required this.index,
  }) : super(key: key);

  final int index;
  @override
```

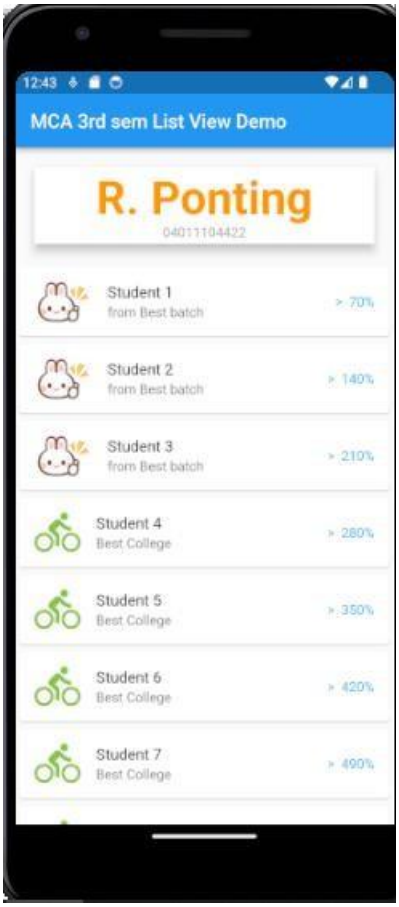
```
  Widget build(BuildContext context) {
    return Card(
      child: ListTile(
      leading: Icon(
          Icons.directions_bike,
          size: 48.0,
          color: Colors.lightGreen,
        ),
        title: Text('Student $index'),
        subtitle: Text('Best College'),
        trailing: Text(
          ' >  ${index * 70}%',
          style: TextStyle(color: Colors.lightBlue),
        ),
        onTap: () {
          print('Tapped on Row $index');
        },
      ),
    );
  }
}
```

RowWithCardWidget.dart:

```
import 'package:flutter/material.dart';


class RowWithCardWidget extends StatelessWidget{
  const RowWithCardWidget({
    Key? key, required this.index,
  }) : super(key: key);

  final int index;
  @override
  Widget build(BuildContext context){
    return Card(
      child: ListTile(
        leading: Image.asset(
          'app_img_src/f0.png', // Adjust the path accordingly
          width: 60,
          height: 60,
        ),
        // leading: Icon(
        //   Icons.flight,
        //   size: 58.0,
        //   color: Colors.lightBlue,
        // ),
        title: Text('Student $index'),
        subtitle: Text('from Best batch'),
        trailing: Text(
```

```
          ' >  ${index * 70}%',
          style: TextStyle(color: Colors.lightBlue),
        ),
        onTap: () {
          print('Tapped on Row $index');
        },
      ),
    );
  }

}
```

Output:



Q27. Create a flutter application using Hero Widget.

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Hero Sample',
      theme: ThemeData(
        primarySwatch: Colors.teal,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: HomePage(),
    );
  }
}
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Hero Sample'),
      ),
      body: Center(
        child: GestureDetector(
          onTap: () {
          Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => DetailPage(),
              ),
            );
          },
          child: Hero(
            tag: 'iconTag',
            child: Container(
              width: 100.0,
              height: 100.0,
              decoration: BoxDecoration(
                color: Colors.blue,
                borderRadius: BorderRadius.circular(16.0),
              ),
              child:  Icon(
                Icons.star,
                size: 50.0,
                color: Colors.white,
```

```
            ),
          ),
        ),
      ),
    );
  }
}
class DetailPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Hero Sample'),
      ),
      body: Center(
        child: Hero(
        tag: 'iconTag',
          child: Container(
            width: 200.0,
            height: 200.0,
            decoration: BoxDecoration(
              color: Colors.blue,
              borderRadius: BorderRadius.circular(16.0),
            ),
            child:   Icon(
              Icons.star,
              size: 100.0,
              color: Colors.white,
            ),
          ),
        ),
      ),
    );
  }
}
```

Output:

Q28. Create a flutter application using Card Widget.

```
import 'dart:math';
import 'package:flutter/material.dart';

void main(){
  runApp(MyApp());
}

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context){
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Mca 3rd - Card Demo',
      home: HomePage(),
    );
  }
}

class HomePage extends StatefulWidget{
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage>{
  @override
  Widget build(BuildContext context){
    return Scaffold(
      appBar: AppBar(
        title: Text('MCA 3rd sem'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(25),
        child: Center(
            child: Column(children: [
          Card(
            elevation: 8.0,
            color: Colors.white,
            margin: EdgeInsets.all(16.0),
            shape: StadiumBorder(),
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: <Widget>[
                Text(
                  'Bond 007',
                  textAlign: TextAlign.center,
                  style: TextStyle(
                    fontWeight: FontWeight.bold,
                    fontSize: 48.0,
                    color: Colors.orange,
```

```
        ),
      ),
      Text('2022-24 batch',
        textAlign: TextAlign.center,
        style: TextStyle(color: Colors.grey),),
      Text('BCIIT',
        textAlign: TextAlign.center,
        style: TextStyle(color: Colors.pinkAccent),)
    ],
  ),
),
Card(
  elevation: 8.0,
  color: Colors.white,
  margin: EdgeInsets.all(16.0),
  shape: OutlineInputBorder(),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Text(
        'John Wick 066',
        textAlign: TextAlign.center,
        style: TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 48.0,
          color: Colors.orange,
        ),
      ),
      Text('2022-24 batch',
        textAlign: TextAlign.center,
        style: TextStyle(color: Colors.grey),),
      Text('BCIIT',
        textAlign: TextAlign.center,
        style: TextStyle(color: Colors.pinkAccent),)
    ],
  ),
),
Card(
  elevation: 8.0,
  color: Colors.white,
  margin: EdgeInsets.all(16.0),
  shape: StarBorder(),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Text(
        'Malinga 033',
        textAlign: TextAlign.center,
        style: TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 48.0,
```

```
                    color: Colors.orange,
                  ),
                ),
                Text('2022-24 batch',
                  textAlign: TextAlign.center,
                  style: TextStyle(color: Colors.grey),),
                Text('BCIIT',
                  textAlign: TextAlign.center,
                  style: TextStyle(color: Colors.pinkAccent),)
              ],
            ),
          )
        ],)
      ),
    ),
  );
 }
}
```

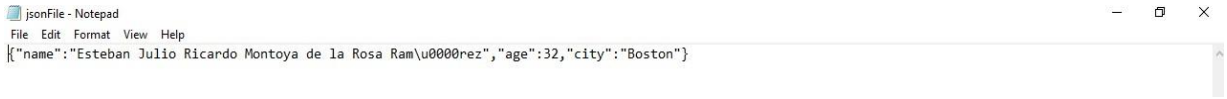
Output:

Q29. Write a dart program to read and write data to a JSON file.

```
import 'dart:convert';
import 'dart:io';

void main() {
  // Path to the JSON file
  String filePath = './jsonFile.json';

  // Take input from the user
  Map<String, dynamic> userData = getUserInput();

  // Write user input to the JSON file
  writeJsonToFile(filePath, userData);

  // Read data from the JSON file
  Map<String, dynamic> dataRead = readJsonFromFile(filePath);

  // Display the read data
  print('Data read from JSON file:');
  print(dataRead);
}

Map<String, dynamic> getUserInput() {
  print('\n\nEnter user data:');
  print('\nName:');
  String name = stdin.readLineSync() ?? '';

  print('\nAge:');
  int age = int.tryParse(stdin.readLineSync() ?? '') ?? 0;

  print('\nCity:');
  String city = stdin.readLineSync() ?? '';

  return {
    'name': name,
    'age': age,
    'city': city,
  };
}
void writeJsonToFile(String filePath, Map<String, dynamic> data) {
  // Convert data to JSON string
  String jsonData = jsonEncode(data);

  // Open the file for writing
  File file = File(filePath);
  file.writeAsStringSync(jsonData);
  print('Data written to JSON file successfully.');

Map<String, dynamic> readJsonFromFile(String filePath) {
  // Check if the file exists
```

```
  File file = File(filePath);
  if (!file.existsSync()) {
    print('File does not exist. Returning an empty map.');
    return {};
  }

  // Read the JSON content from the file
  String jsonContent = file.readAsStringSync();

  // Parse the JSON string into a Map
  Map<String, dynamic> jsonData = jsonDecode(jsonContent);

  return jsonData;
}
```

Output:



Q30. Create a flutter application using Grid view widget and hero widget.

main.dart:

```
import 'package:flutter/material.dart';

import 'GridClassBuild.dart';

void main() => runApp(GridApp());

class GridApp extends StatelessWidget {
  const GridApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
      appBar: AppBar(
          title: Text('Image Grid'),
        ),
        body: GridViewBuilderWidget(),
      ),
    );
  }
}
```

GridClassBuild.dart:

```
import 'package:flutter/material.dart';

class GridIcons {
  static List<int> getIconList() {
    return List<int>.generate(8, (index) => index);
  }
}

class GridViewBuilderWidget extends StatelessWidget {
  const GridViewBuilderWidget({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    List<int> _iconList = GridIcons.getIconList();
    return GridView.builder(
      itemCount: _iconList.length,
      padding: EdgeInsets.all(8.0),
      gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(
        maxCrossAxisExtent: 150.0,
      ),
      itemBuilder: (BuildContext context, int index) {
        print('_buildGridViewBuilder $index');
```

```
        return Card(
          color: Colors.lightGreen.shade50,
          margin: EdgeInsets.all(8.0),
          child: InkWell(
            child: Hero(
              tag: 'hero-rectangle',
              child: BoxWidget(size: Size(50.0, 50.0), imgIndex: index),
            ),
            onTap: () {
              _ZoomImgPage(context, index);
              print('Tapped on index $index');
            },
          ),
        );
      },
    );
  }
}

void _ZoomImgPage(BuildContext context, int index) {
  Navigator.of(context).push(MaterialPageRoute<void>(
  builder: (BuildContext context) => Scaffold(
      appBar: AppBar(
        title: const Text('Image Zoom'),
      ),
      body: Center(
        child: InkWell(
        child: Hero(
            tag: 'hero-rectangle',
            child: BoxWidget(size: Size(50.0, 50.0), imgIndex: index),
          ),
          onTap: () {
            Navigator.pop(context); // This will pop the route and go back
          },
        ),
      ),
    ),
  ));
}

class BoxWidget extends StatelessWidget {
  BoxWidget({Key? key, required this.size, required this.imgIndex})
      : super(key: key);

  final Size size;
  final int imgIndex;

  @override
  Widget build(BuildContext context) {
    return Container(
        width: size.width+200,
```

```
        height: size.height+200,
        color: Colors.blue,
        child: Image.asset('app_img_src/f$imgIndex.png'));
  }
}
```

Output: