NAME: MUHAMMAD IBRAHIM SHAIKH.
ROLL NO: 203304.
COURSE: FLUTTER.
INSTRUCTOR: BILAL REHMAN.
ASSIGNMENT NO : 04

Write a program that takes a list of numbers as input and prints the
even numbers in the list using a for loop.
Example:
Input: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
void main() {
List<int> numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
  print("Even numbers in the list:");
  for (int number in numbers) {
    if (number % 2 == 0) {
      print(number);
    }
  }
}
```

Write a program that prints the Fibonacci sequence up to a given
number using a for loop.
Example:
Input: 10

```
void main() {
  int limit = 10;
  int first = 0;
  int second = 1;
  int next;
  print("Fibonacci sequence up to $limit:");
```

```
  for (int i = 0; i < limit; i++) {
    if (i <= 1) {
      next = i;
    } else {
      next = first + second;
      first = second;
      second = next;}
    print(next);
  }
}
```

Implement a code that checks whether a given number is prime or not.
Example:
Input: 17

```
void main() {
  int number =17;
  bool isPrime = checkPrime(number);
  if (isPrime) {
   print("$number is a prime number.");
  } else {
    print("$number is not a prime number.");
  }
}
bool checkPrime(int n) {
  if (n <= 1) {
    return false;
  }
```

```
  for (int i = 2; i <= n / 2; i++) {
    if (n % i == 0) {
      return false;
    }
  }
  return true;
}
```

Implement a code that finds the factorial of a number using a while
loop or for loop.
Example:
Input: 5

```
void main() {
  int number = 5;
  int factorial = findFactorial(number);
  print("Factorial of $number is $factorial");
}
int findFactorial(int n) {
  int factorial = 1;
  for (int i = 1; i <= n; i++) {
    factorial *= i;
  }
  return factorial;
}
```

Write a program that calculates the sum of all the digits in a given

number using a while loop.
Example:
Input: 12345

```
void main() {
  int number = 12345;
  int sum = calculateDigitSum(number);
  print("Sum of digits in $number is $sum");
}
int calculateDigitSum(int n) {
  int sum = 0;
  while (n != 0) {
    sum += n % 10;
    n ~/= 10;
  }
  return sum;
}
```

Implement a code that finds the largest element in a list using a for loop.
Example:
Input: [3, 9, 1, 6, 4, 2, 8, 5, 7]

```
void main() {
  List<int> numbers = [3, 9, 1, 6, 4, 2, 8, 5, 7];
  int largest = findLargestElement(numbers);
  print("The largest element in the list is: $largest");
}
int findLargestElement(List<int> list) {
```

```
  if (list.isEmpty) {
    throw ArgumentError("List is empty");
  }
  int largest = list[0]; // Initialize the largest with the first
element
  for (int i = 1; i < list.length; i++) {
    if (list[i] > largest) {
      largest = list[i]; // Update largest if a larger element is
found
    }
  }
  return largest;
}
```

Write a program that prints the multiplication table of a given number
using a for loop.
Example:
Input: 5

```
void main() {
  int number = 5;
  int tableSize = 10;
  print("Multiplication table of $number:");
  for (int i = 1; i <= tableSize; i++) {
    print("$number x $i = ${number * i}");
  }
}
```

Implement a function that checks if a given string is a palindrome.
Example:
Input: "radar"

```
bool isPalindrome(String str) {
  int start = 0;
  int end = str.length - 1;
  while (start < end) {
    if (str[start] != str[end]) {
      return false;
    }
    start++;
    end--;
  }
  return true;
}
void main() {
  String input = "radar";
  if (isPalindrome(input)) {
    print("$input is a palindrome.");
  } else {
    print("$input is not a palindrome.");
  }
}
```

Write a program to display the cube of the number up to an integer.
Test Data :

Input number of terms : 5

```
void main() {
  stdout.write("Input number of terms: ");
  int numberOfTerms = int.parse(stdin.readLineSync()!);
  print("Cube of the numbers up to $numberOfTerms:");
  for (int i = 1; i <= numberOfTerms; i++) {
    int cube = i * i * i;
    print("Number: $i, Cube: $cube");
  }
}
```

Write a program to display a pattern like a right angle triangle using an
asterisk using loop.
The pattern like :

```
*
**
***
****
```

```
void main() {
  int rows = 4; // Change this to adjust the number of rows in
the triangle
  print("Pattern:");
  for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= i; j++) {
      // Print asterisks based on the row number
      print("*", end: "");}
```

```
    // Move to the next line after printing asterisks for each row
    print("");
  }
}
```

Write a program to display a pattern like a right angle triangle with a number using loop.

The pattern like :

1
12
123
1234

```
void main() {
  int rows = 4; // Change this to adjust the number of rows in the triangle
  print("Pattern:");
  for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= i; j++) {
      // Print numbers from 1 to the current column number
      print("$j", end: "");
    }
   // Move to the next line after printing numbers for each row
    print("");
  }
}
```

Write a program to make such a pattern like a right angle triangle with
a number which will repeat a number in a row.
The pattern like :
1
22
333
4444

```
void main() {
  int rows = 4; // Change this to adjust the number of rows in the triangle
  print("Pattern:");
  for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= i; j++) {
      // Print the current row number i times
      print("$i", end: "");
    }
    // Move to the next line after printing numbers for each row
    print("");
  }
}
```

Write a program to make such a pattern like a right angle triangle with the number increased by 1 using loop.. The pattern like :
1
2 3
4 5 6

7 8 9 10

```
void main() {
  int rows = 4; // Change this to adjust the number of rows in the triangle
  int number = 1; // Initialize the starting number
  print("Pattern:");
  for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= i; j++) {
      // Print the current number and increment it
      print("$number ", end: "");
      number++;
    }
    // Move to the next line after printing numbers for each row
    print("");
  }
}
```

Write a program to make a pyramid pattern with numbers increased by

1
2 3
4 5 6
7 8 9 10

```
void main() {
  int rows = 4; // Change this to adjust the number of rows in the pyramid
  int number = 1; // Initialize the starting number
```

```
  print("Pattern:");
  for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= rows - i; j++) {
      // Print spaces to align the numbers correctly
      print(" ", end: "");
    }
    for (int k = 1; k <= i; k++) {
      // Print the current number and then increment it
      print("$number ", end: "");
      number++;
    }
    // Move to the next line after printing numbers for each row
    print("");
  }
}
```

Write a program to make such a pattern as a pyramid with an asterisk.

```
*
* *
* * *
* * * *
void main() {
  int rows = 4; // Change this to adjust the number of rows in the pyramid

  print("Pattern:");
  for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= i; j++) {
```

```
      // Print asterisks based on the current row number
      print("* ", end: "");
    }
    // Move to the next line after printing asterisks for each row
    print("");
  }
}
```

Write a program that asks the user for their email and password. If the email and password match a predefined set of credentials, print "User login successful." Otherwise, keep asking for the email and password until the correct credentials are provided.

```
void main() {
  // Predefined set of credentials
  Map<String, String> credentials = {
    "user@example.com": "password123",
    "admin@example.com": "admin123",
    // Add more credentials if needed
  };
  bool isLoggedIn = false;
  while (!isLoggedIn) {
    // Ask the user for their email and password
    print("Enter your email:");
    String email = readLineSync();
    print("Enter your password:");
    String password = readLineSync();
```

```
    // Check if the provided credentials match any in the
predefined set
    if (credentials.containsKey(email) && credentials[email]
== password) {
      print("User login successful.");
      isLoggedIn = true;
    } else {
      print("Incorrect email or password. Please try again.");
    }
  }
}

// Function to read input from the console
String readLineSync() {
  return stdin.readLineSync()!;
}
```

Write a program that asks the user for their email and password. You are given a list of predefined user credentials (email and password combinations). If the entered email and password match any of the credentials in the list, print "User login successful." Otherwise, keep asking for the email and password until the correct credentials are provided.

```
void main() {
  // Predefined list of user credentials
  List<Map<String, String>> userCredentials = [
```

```
    {"email": "user1@example.com", "password":
"password1"},
    {"email": "user2@example.com", "password":
"password2"},
    // Add more credentials if needed
  ];

  bool isLoggedIn = false;

  while (!isLoggedIn) {
    // Ask the user for their email and password
    print("Enter your email:");
    String email = readLineSync();
    print("Enter your password:");
    String password = readLineSync();

    // Check if the provided email and password match any in
the predefined list
    for (var credentials in userCredentials) {
      if (credentials["email"] == email &&
credentials["password"] == password) {
        print("User login successful.");
        isLoggedIn = true;
        break;
      }
    }

    if (!isLoggedIn) {
      print("Incorrect email or password. Please try again.");
```

```
    }
  }
}

// Function to read input from the console
String readLineSync() {
  return stdin.readLineSync()!;
}
```

Write a program that takes a list of numbers as input and prints the numbers greater than 5 using a for loop and if-else condition.

```
void main() {
  List<int> numbers = [1, 6, 3, 8, 2, 7, 9, 4]; // Example list of numbers

  print("Numbers greater than 5:");

  for (int number in numbers) {
    if (number > 5) {
      print(number);
    }
  }
}
```

Write a program that counts the number of vowels in a given string using a for loop and if-else condition.

```
void main() {
  String input = "Hello World"; // Example input string
  int vowelCount = 0;

  for (int i = 0; i < input.length; i++) {
    if (isVowel(input[i])) {
      vowelCount++;
    }
  }

  print("Number of vowels in the string: $vowelCount");
}

bool isVowel(String letter) {
  String vowels = "aeiouAEIOU";
  return vowels.contains(letter);
}
```

Implement a code that finds the maximum and minimum elements in a list using a for loop and if-else condition.

```
void main() {
  List<int> numbers = [5, 2, 8, 1, 6, 3, 9, 4]; // Example list of numbers

  // Initialize max and min with the first element of the list
  int max = numbers[0];
  int min = numbers[0];
```

```
  // Iterate through the list to find max and min
  for (int i = 1; i < numbers.length; i++) {
    if (numbers[i] > max) {
      max = numbers[i]; // Update max if current element is
greater
    }
    if (numbers[i] < min) {
      min = numbers[i]; // Update min if current element is
smaller
    }
  }

  print("Maximum element: $max");
  print("Minimum element: $min");
}
```

Write a program that calculates the sum of the squares of all odd numbers in a given list using a for loop and if-else condition.

```
void main() {
  List<int> numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]; // Example list
of numbers
  int sumOfOddSquares = 0;

  for (int number in numbers) {
    if (number % 2 != 0) { // Check if the number is odd
      sumOfOddSquares += (number * number); // Add the
square of the odd number to the sum
```

```
    }
  }

  print("Sum of the squares of all odd numbers:
$sumOfOddSquares");
}
```

Write a program that takes a list of student details as input, where each student is represented by a map containing their name, marks, section, and roll number. The program should determine the grade of each student based on their average score (assuming maximum marks for each subject is 100) and print the student's name along with their grade.

List<Map<String, dynamic>> studentDetails = [
{'name': 'John', 'marks': [80, 75, 90], 'section': 'A', 'rollNumber': 101},
{'name': 'Emma', 'marks': [95, 92, 88], 'section': 'B', 'rollNumber': 102},
{'name': 'Ryan', 'marks': [70, 65, 75], 'section': 'A', 'rollNumber': 103},];

```
void main() {
  List<Map<String, dynamic>> studentDetails = [
    {'name': 'John', 'marks': [80, 75, 90], 'section': 'A',
'rollNumber': 101},
    {'name': 'Emma', 'marks': [95, 92, 88], 'section': 'B',
'rollNumber': 102},
```

```
    {'name': 'Ryan', 'marks': [70, 65, 75], 'section': 'A',
'rollNumber': 103},
  ];

  for (var student in studentDetails) {
    String name = student['name'];
    List<int> marks = student['marks'];
    double average = marks.reduce((a, b) => a + b) /
marks.length;

    String grade = calculateGrade(average);

    print("$name - Grade: $grade");
  }
}

String calculateGrade(double average) {
  if (average >= 90) {
    return 'A';
  } else if (average >= 80) {
    return 'B';
  } else if (average >= 70) {
    return 'C';
  } else if (average >= 60) {
    return 'D';
  } else {
    return 'F';
  }
}
```

Implement a code that finds the average of all the negative numbers in a list using a for loop and if-else condition.

```
void main() {
  List<int> numbers = [-5, 10, -3, 8, -7, 4]; // Example list of
numbers
  int sumOfNegatives = 0;
  int countOfNegatives = 0;

  for (int number in numbers) {
    if (number < 0) { // Check if the number is negative
      sumOfNegatives += number; // Add the negative number
to the sum
      countOfNegatives++; // Increment the count of negative
numbers
    }
  }

  double averageOfNegatives = countOfNegatives > 0 ?
sumOfNegatives / countOfNegatives : 0;

  print("Average of all negative numbers:
$averageOfNegatives");
}
```

Write a program that takes a list of integers as input and returns a new list containing only the prime numbers from the original list. Implement the solution using a for loop and logical operations.

Input: [4, 7, 10, 13, 16, 19, 22, 25, 28, 31]

```
void main() {
  List<int> numbers = [4, 7, 10, 13, 16, 19, 22, 25, 28, 31]; //
Example list of numbers
  List<int> primeNumbers = [];

  for (int number in numbers) {
    if (isPrime(number)) {
      primeNumbers.add(number);
    }
  }

  print("Original list: $numbers");
  print("Prime numbers in the list: $primeNumbers");
}

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