**Assignment #2**

Q.1: Create a list of names and print all names using the List method.

**CODE:**

void main() {

  // Create a list of names

  List<String> names = ["Muhib", "Mallick", "Hassan", "Bilal", "Hannan"];

  // Print all names using forEach method

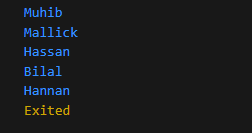
  names.forEach((name) {

    print(name);

  });

}

**OUTPUT:**

****

Q.2: Create an empty list of type string called days. Use the add method to add names of 7 days and print all days.

**CODE:**

void main() {

  // Create an empty list of type String

  List<String> days = [];

  // Add names of the 7 days using add method

  days.add("Monday");

  days.add("Tuesday");

  days.add("Wednesday");

  days.add("Thursday");

  days.add("Friday");

  days.add("Saturday");

  days.add("Sunday");

  // Print all days using forEach method

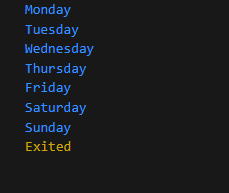
  days.forEach((day) {

    print(day);

  });

}

**OUTPUT:**

****

Q.3: Create a list of Days and remove one by one from the end of list.

**CODE:**

void main() {

  // Create a list of days

  List<String> days = [

    "Monday",

    "Tuesday",

    "Wednesday",

    "Thursday",

    "Friday",

    "Saturday",

    "Sunday"

  ];

  // Remove days one by one from the end of the list

  while (days.isNotEmpty) {

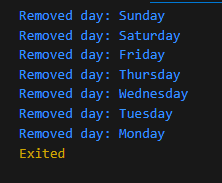
    String removedDay = days.removeLast();

    print("Removed day: $removedDay");

  }

}

**OUTPUT:**

****

Q.4: Create a list of numbers and create one empty list, now check for every index number is EVEN or ODD.

if number is even then add true into empty list and if number is odd then add false into empty list,

both list needs to print at the end.

**CODE:**

void main() {

  List<int> numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

  List<bool> evenOddList = [];

  numbers.forEach((number) {

    bool isEven = number % 2 == 0;

    evenOddList.add(isEven);

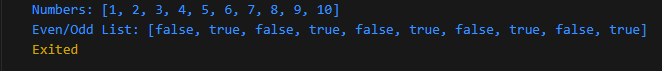
  });

  print('Numbers: $numbers');

  print('Even/Odd List: $evenOddList');

}

**OUTPUT:**

****

Q.5 Create a map with name, phone keys and store some values to it.

// Use where to find all keys that have length 4.

**CODE:**

void main() {

void main() {

  // Create a map with name and phone keys

  Map<String, dynamic> contactMap = {

    "name": "Muhib",

    "phone": 030178601,

    "city": "Karachi",

    "address": "Karachi Sindh, Pakistan"

  };

  // Find keys with length 4 using where method

  List<String> keysWithLength4 =

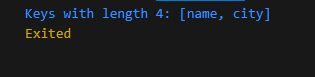
      contactMap.keys.where((key) => key.length == 4).toList();

  // Print the keys with length 4

  print("Keys with length 4: $keysWithLength4");

}

**OUTPUT:**

****

Q.6: Create Map variable name world then inside it create countries Map,

Key will be the name country & country value will have another map having capital City-, currency and language to it.

by using any country key print all the value of Capital & Currency.

\*/

**CODE:**

import 'dart:io';

void main() {

  // Create the world map

  Map<String, Map<String, String>> world = {

    "USA": {

      "capitalCity": "Washington D.C.",

      "currency": "US Dollar",

      "language": "English"

    },

    "Germany": {

      "capitalCity": "Berlin",

      "currency": "Euro",

      "language": "German"

    },

    "Japan": {

      "capitalCity": "Tokyo",

      "currency": "Japanese Yen",

      "language": "Japanese"

    }

  };

  // Print the capital and currency of a specific country

  String countryKey = "USA";

  print(world[countryKey]);

  exit;

  var countryInfo = world[countryKey];

  if (countryInfo != null) {

    var capitalCity = countryInfo["capitalCity"];

    var currency = countryInfo["currency"];

    print("Country: $countryKey");

    print("Capital City: $capitalCity");

    print("Currency: $currency");

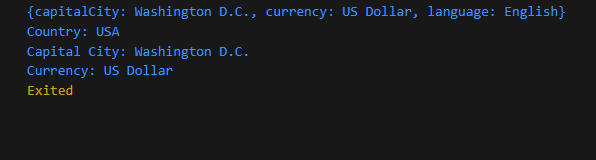
  } else {

    print("Country not found!");

  }

}

**OUTPUT:**

****

Q.7: Q.7:

Map<String, double> expenses = {

'sun': 3000.0,

'mon': 3000.0,

'tue': 3234.0,

};

Check if "fri" exist in expanses;

if exist change it's value to 5000.0 otherwise add 'fri' to expenses

and set its value to 5000.0 then print expenses.

**CODE:**

void main() {

  Map<String, double> expenses = {

    'sun': 3000.0,

    'mon': 3000.0,

    'tue': 3234.0,

  };

  // Check if "fri" exists in expenses

  if (expenses.containsKey('fri')) {

    expenses['fri'] = 5000.0;

  } else {

    expenses['fri'] = 5000.0;

  }

  // Print expenses

  print(expenses);

}

**OUTPUT:**

****

Q.8: remove all false values from below list by using removeWhere or retainWhere property.

List<Map<String, bool>> usersEligibility = [

{'name': 'John', 'eligible': true},

{'name': 'Alice', 'eligible': false},

{'name': 'Mike', 'eligible': true},

{'name': 'Sarah', 'eligible': true},

{'name': 'Tom', 'eligible': false},

];**CODE:**

void main() {

  List<Map<String, dynamic>> usersEligibility1 = [

    {'name': 'John', 'eligible': true},

    {'name': 'Alice', 'eligible': false},

    {'name': 'Mike', 'eligible': true},

    {'name': 'Sarah', 'eligible': true},

    {'name': 'Tom', 'eligible': false},

  ];

  // Remove map entries with 'false' values for 'eligible' key

  usersEligibility1.removeWhere((user) => user['eligible'] == false);

  // Print the updated usersEligibility list

  // print(usersEligibility1);

  List<Map<String, dynamic>> usersEligibility = [

    {'name': 'John', 'eligible': true},

    {'name': 'Alice', 'eligible': false},

    {'name': 'Mike', 'eligible': true},

    {'name': 'Sarah', 'eligible': true},

    {'name': 'Tom', 'eligible': false},

  ];

  // Remove map entries with 'false' values for 'eligible' key

  usersEligibility.retainWhere((user) => user['eligible'] == true);

  // Print the updated usersEligibility list

  print(usersEligibility);

}

**OUTPUT:**

****

Q9: Given a list of integers, write a dart code that returns the maximum value from the list.

**CODE:**

void main() {

  List<int> numbers = [10, 5, 8, 2, 20, 134, 15, 3];

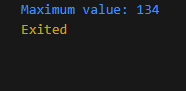
  int maxValue =

      numbers.reduce((value, element) => value > element ? value : element);

  print("Maximum value: $maxValue");

}

**OUTPUT:**

****

Q10 Write a Dart code that takes in a list of strings and removes any duplicate elements, returning a new list without duplicates. The order of elements in the new list should be the same as in the original list.

**CODE:**

void main() {

  List<String> originalList = [

    'apple',

    'banana',

    'orange',

    'banana',

    'grape',

    'apple'

  ];

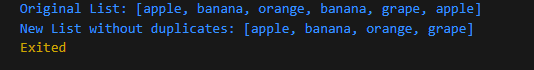
  List<String> newList = originalList.toSet().toList();

  print("Original List: $originalList");

  print("New List without duplicates: $newList");

}

**OUTPUT:**

****

Q11 11: Write a Dart code that takes in a list and an integer n as parameters.

The program should print a new list containing the first n elements from the original list.

**CODE:**

void main() {

  List<int> originalList = [1, 2, 3, 4, 5, 6, 7];

  int n = 3;

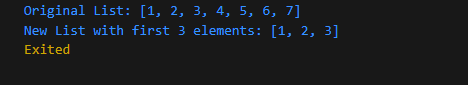
  List<int> newList = originalList.sublist(0, n);

  print("Original List: $originalList");

  print("New List with first $n elements: $newList");

}

**OUTPUT:**

****

Q12 Write a Dart code that takes in a list of strings and prints a new list with the elements in reverse order.

The original list should remain unchanged.

**CODE:**

void main() {

  List<String> originalList = ['apple', 'banana', 'orange', 'grape'];

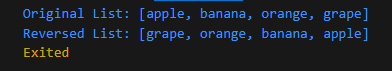
  List<String> reversedList = List.from(originalList.reversed);

  print("Original List: $originalList");

  print("Reversed List: $reversedList");

}

**OUTPUT:**

****

Q13

**CODE:**

void main() {

  List<int> originalList = [1, 2, 3, 2, 4, 3, 5, 1];

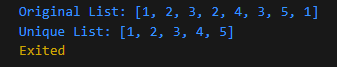
  List<int> uniqueList = originalList.toSet().toList();

  print("Original List: $originalList");

  print("Unique List: $uniqueList");

}

**OUTPUT:**

****

Q14

**CODE:**

void main() {

  List<int> originalList = [5, 2, 7, 1, 4, 3];

  List<int> sortedList = List.from(originalList);

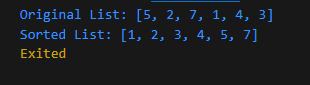
  sortedList.sort();

  print("Original List: $originalList");

  print("Sorted List: $sortedList");

}

**OUTPUT:**

****

Q15

**CODE:**

void main() {

  List<int> originalList = [-2, 5, -9, 3, -1, 7, -4];

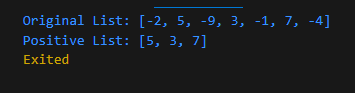
  List<int> positiveList = originalList.where((number) => number > 0).toList();

  print("Original List: $originalList");

  print("Positive List: $positiveList");

}

**OUTPUT:**

****

Q16

**CODE:**

void main() {

  List<int> originalList = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

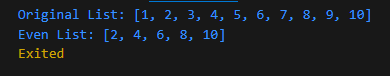
  List<int> evenList = originalList.where((number) => number % 2 == 0).toList();

  print("Original List: $originalList");

  print("Even List: $evenList");

}

**OUTPUT:**

****

Q17

**CODE:**

void main() {

  List<int> originalList = [1, 2, 3, 4, 5];

  List<int> squaredList =

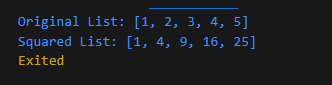
      originalList.map((number) => number \* number).toList();

  print("Original List: $originalList");

  print("Squared List: $squaredList");

}

**OUTPUT:**

****

Q18

**CODE:**

void main() {

  Map<String, dynamic> person = {

    'name': 'Saqib',

    'age': 17,

    'isStudent': true,

  };

  bool isStudent = person['isStudent'];

  int age = person['age'];

  if (isStudent && age > 18) {

    print("Eligible");

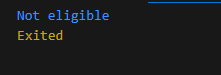
  } else {

    print("Not eligible");

  }

}

**OUTPUT:**

****

Q19

**CODE:**

void main() {

  Map<String, dynamic> product = {

    'name': 'Orange',

    'price': 1.99,

    'quantity': 0,

  };

  int quantity = product['quantity'];

  if (quantity > 0) {

    print("In stock");

  } else {

    print("Out of stock");

  }

}

**OUTPUT:**

****

Q20

**CODE:**

void main() {

  Map<String, dynamic> car = {

    'brand': 'Kia',

    'color': 'Black',

    'isSedan': true,

  };

  bool isSedan = car['isSedan'];

  String color = car['color'];

  if (isSedan && color == 'Black') {

    print("Match");

  } else {

    print("No match");

  }

}

**OUTPUT:**

****

Q21

**CODE:**

void main() {

  Map<String, dynamic> user = {

    'name': 'Saqib Ishfaque',

    'isAdmin': true,

    'isActive': true,

  };

  bool isAdmin = user['isAdmin'];

  bool isActive = user['isActive'];

  if (isAdmin && isActive) {

    print("Active admin");

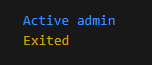
  } else {

    print("Not an active admin");

  }

}

**OUTPUT:**

****

Q22

**CODE:**

void main() {

  Map<String, int> shoppingCart = {

    'Banana': 2,

    'Orange': 3,

    'Apple': 5,

    'Mango': 1,

  };

  String productName = 'Orange';

  if (shoppingCart.containsKey(productName)) {

    print("Product found");

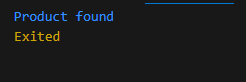
  } else {

    print("Product not found");

  }

}

**OUTPUT:**

****