**Flutter Assignment-3**

Hitesh Walia, MCA

00311104422

1. **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';

void main() {

List<int> factors = [];

stdout.write("Enter any number: ");

int? n = int.parse(stdin.readLineSync() ?? " ");

for (int i = 1; i <= n; i++) {

if (n % i == 0) {

factors.add(i);

}

}

print('Factors of $n are: $factors');

}

1. **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() {

stdout.write("Enter any number: ");

int n = int.parse(stdin.readLineSync() ?? " ");

List<int> primeFactors = [];

for (int factor = 2; factor <= n; factor++) {

while (n % factor == 0) {

primeFactors.add(factor);

n ~/= factor;

}

}

print('Prime factors of $n are: $primeFactors');

}

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

import 'dart:io';

void main() {

int count=0;

stdout.write("Enter any number: ");

int? n = int.parse(stdin.readLineSync() ?? " ");

for (int i = 1; i <= n; i++) {

if (n % i == 0) {

count++;

}

}

print('Number of factors of $n are: $count');

}

1. **Write a program to reverse the numbers given in a list**

void main() {

List<int> numbers = [123, 456, 789, 321];

List<int> reversedNumbers = reverseList(numbers);

print('Original Numbers: $numbers');

print('Reversed Numbers: $reversedNumbers');

}

List<int> reverseList(List<int> numbers) {

List<int> rev = [];

for (int i in numbers) {

int reversedNumber = reverseNumber(i);

rev.add(reversedNumber);

}

return rev;

}

int reverseNumber(int number) {

int reversedNumber = 0;

while (number > 0) {

int digit = number % 10;

reversedNumber = reversedNumber \* 10 + digit;

number ~/= 10;

}

return reversedNumber;

}

1. **Write a Dart program that checks if a specified number appears in**

**every pair of adjacent integers of a given array of integers.**

void main() {

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

int target = 3; bool foundInPairs = checkInAdjacentPairs(numbers, target);

if (foundInPairs) {

print('$target appears in every pair of adjacent integers.');

} else {

print('$target does not appear in every pair of adjacent integers.');

}

}

bool checkInAdjacentPairs(List<int> numbers, int target) {

for (int i = 0; i < numbers.length - 1; i++) {

if (numbers[i] != target && numbers[i + 1] != target) {

return false;

}

}

return true;

}

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

**Example**

**array1 = [1,3,4,2]**

**array2 = [8,2,7, 5]**

**result = [1,2,2,3,4,5,7,8]**

void main() {

List<int> array1 = [1, 3, 4, 2];

List<int> array2 = [8, 2, 7, 5];

List<int> result = mergeAndSortArrays(array1, array2);

print('Result: $result');

}

List<int> mergeAndSortArrays(List<int> array1, List<int> array2) {

List<int> mergedArray = [...array1, ...array2];

mergedArray.sort();

return mergedArray;

}

1. **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;

Mobile(this.brand, this.color, this.camera);

void printDetails() {

print('Brand: $brand');

print('Color: $color');

print('Camera: $camera MP');

}

}

void main() {

Mobile mobile1 = Mobile('Samsung', 'Black', 16.0);

Mobile mobile2 = Mobile('Apple', 'Silver', 12.0);

Mobile mobile3 = Mobile('Google Pixel', 'White', 20.0);

print('Mobile 1 Details:');

mobile1.printDetails();

print('\nMobile 2 Details:');

mobile2.printDetails();

print('\nMobile 3 Details:');

mobile3.printDetails();

}

1. **Write a program to print number into words**

**Input N=125**

**Output : One Two Five**

void main() {

int number = 125; // Replace with the number you want to convert

print('Input N = $number');

String words = numberToWords(number);

print('Output: $words');

}

String numberToWords(int number) {

if (number == 0) {

return 'Zero';

}

final List<String> unitWords = [ 'One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine'];

final List<String> tensWords = ['', '', 'Twenty', 'Thirty', 'Forty', 'Fifty', 'Sixty', 'Seventy', 'Eighty', 'Ninety'];

final List<String> teensWords = [ 'Ten', 'Eleven', 'Twelve', 'Thirteen', 'Fourteen', 'Fifteen', 'Sixteen', 'Seventeen', 'Eighteen', 'Nineteen'];

String result = '';

int thousands = number ~/ 1000;

if (thousands > 0) {

result += '${unitWords[thousands]} Thousand ';

number %= 1000;

}

int hundreds = number ~/ 100;

if (hundreds > 0) {

result += '${unitWords[hundreds]} Hundred ';

number %= 100;

}

if (number >= 20) {

int tens = number ~/ 10;

result += '${tensWords[tens]} ';

number %= 10;

}

else if (number >= 10) {

result += '${teensWords[number - 10]}';

number = 0; // We've already handled the tens and ones place

}

if (number > 0) {

result += '${unitWords[number]}';

}

return result.trim();

}

1. **Write a program to print the number into words**

**Input N=125**

**Output: One hundred twenty-five**

void main() {

int number = 125;

print('Input N = $number');

String words = numberToWords(number);

print('Output: $words');

}

String numberToWords(int number) {

if (number == 0) {

return 'Zero';

}

final List<String> unitWords = ['', 'One', 'Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine'];

final List<String> tensWords = ['', '', 'Twenty', 'Thirty', 'Forty', 'Fifty', 'Sixty', 'Seventy', 'Eighty', 'Ninety'];

final List<String> teensWords = ['Ten', 'Eleven', 'Twelve', 'Thirteen', 'Fourteen', 'Fifteen', 'Sixteen', 'Seventeen', 'Eighteen', 'Nineteen’];

String result = '';

int thousands = number ~/ 1000;

if (thousands > 0) {

result += '${unitWords[thousands]} Thousand ';

number %= 1000;

}

int hundreds = number ~/ 100;

if (hundreds > 0) {

result += '${unitWords[hundreds]} Hundred ';

number %= 100;

}

if (number >= 20) {

int tens = number ~/ 10;

result += '${tensWords[tens]} ';

number %= 10;

} else if (number >= 10) {

result += '${teensWords[number - 10]}';

number = 0; // We've already handled the tens and ones place

}

if (number > 0) {

result += '${unitWords[number]}';

}

return result.trim();

1. **Write a program to print binary equivalent of a number**

**Input : 8**

**Output : 100**

void main() {

int decimalNumber = 8; String binaryEquivalent = decimalToBinary(decimalNumber);

print('Input: $decimalNumber');

print('Output: $binaryEquivalent');

}

String decimalToBinary(int decimalNumber) {

if (decimalNumber == 0) {

return '0';

}

String binary = '';

while (decimalNumber > 0) {

int remainder = decimalNumber % 2;

binary = '$remainder$binary';

decimalNumber ~/= 2;

}

return binary;

}