LAB-1

1.

import 'package:hello\_dart/hello\_dart.dart' as hello\_dart;

void main() {

print('Hello, World!');

}

PS D:\CE\_116\hello\_dart> dart run bin/hello\_dart.dart

Hello, World!



2.

import 'package:fact/fact.dart' as fact;

void main() {

int number = 5;

int factorial = 1;

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

factorial \*= i;

}

print('Factorial of $number is $factorial');

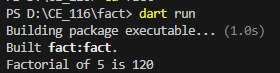
}

PS D:\CE\_116\fact> dart run

Building package executable... (1.0s)

Built fact:fact.

Factorial of 5 is 120



3.

import 'package:prime/prime.dart' as prime;

void main() {

int start = 10;

int end = 50;

print('Prime numbers between $start and $end are:');

for (int num = start; num <= end; num++) {

if (isPrime(num)) {

print(num);

}

}

}

bool isPrime(int n) {

if (n <= 1) return false;

for (int i = 2; i <= n ~/ 2; i++) {

if (n % i == 0) return false;

}

return true;

}

PS D:\CE\_116\prime> dart run

Building package executable...

Built prime:prime.

Prime numbers between 10 and 50 are:

11

13

17

19

23

29

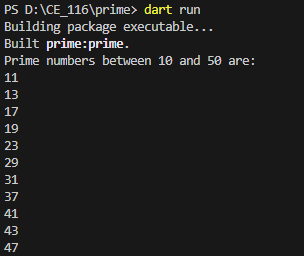
31

37

41

43

47



4.

import 'package:armstrong/armstrong.dart' as armstrong;

import 'dart:math';

void main() {

int start = 1;

int end = 9999;

print('Armstrong numbers between $start and $end are:');

for (int num = start; num <= end; num++) {

if (isArmstrong(num)) {

print(num);

}

}

}

bool isArmstrong(int num) {

int sum = 0;

int original = num;

int digits = num.toString().length;

while (num > 0) {

int digit = num % 10;

sum += pow(digit, digits).toInt();

num ~/= 10;

}

return sum == original;

}

PS D:\CE\_116\armstrong> dart run

Building package executable...

Built armstrong:armstrong.

Armstrong numbers between 1 and 9999 are:

1

2

3

4

5

6

7

8

9

153

370

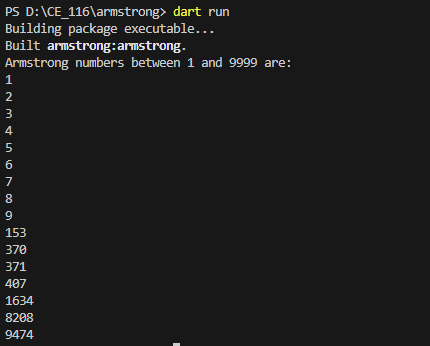
371

407

1634

8208

9474



5.

import 'package:pattern/pattern.dart' as pattern;

void main() {

int rows = 5;

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

String stars = '\*' \* i;

print(stars);

}

}

PS D:\CE\_116\pattern> dart run

Building package executable...

Built pattern:pattern.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

