Mobile Development Laboratory Week 2: Dart Programming		
Name: Worawut Khumnoi	ID: 6431501102	Section: 4
Date: 27/1/2023	Due: Next Wednesday midnight	

Objectives

- To practice Dart programming
- To experiment with Dart syntaxes

References

https://dart-tutorial.com/

To get a console's input:

```
import 'dart:io';

void main() {
  // write without new line
  stdout.write('Name: ');
  // get a nullable string
  String? name = stdin.readLineSync();
  // show the output
  print('Your name is $name');
}
```

To run the above file in the terminal, assume that the filename is 'example.dart':

```
dart run example.dart
```

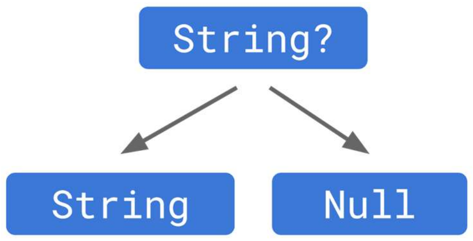
Null safety

By default, all dart's variables are non-nullable or cannot be null. Once we create a normal variable, we usually cannot let it to be empty or null. Note that empty (void) and null are not the same.

```
  String name;    //warning: must be initialized
  String address = null;  //error: a varialbe can't be null
```

If we really know that a variable can have both null value and normal value, we must use "?" operator in the variable declaration. Then this variable can keep either null or normal data type.

```
  String? name;     //nullable String variable
  name = null;      //can be null
  name = 'Mike';    //can keep String
```



The example of using nullable variable is when we get the console's input using "readLineSync()".

```
void main() {
  stdout.write('Enter a name: ');
  String? name = null;
  //get keyboard's input, can be null
  name = stdin.readLineSync();
  print(name);
}
```

If you want to force convert nullable variable to normal variable, you can use null assertion operator "!".

```
  String? name;
  //String name2 = name;    // error: can't assign String? to String
  String name2 = name!;     // OK, we use !
```

If we want to use a property or a method of nullable variable, we can use null aware operator '?.'

```
  String? name;
  name = null;
  //print(name.length);   // compile error because name is null
  print(name?.length);    // OK, will show null if name is null
  //print(name!.length);    // runtime error because name is null

  name = 'Tom';
  print(name.length);     // now, name is String, no need to use ? or !
```

Before using nullable variable, we should check whether it is null. If it is not, we do not need to use null assertion operator “!”.

```
import 'dart:io';

void main() {
  int age = 0;
  stdout.write('Input age: ');
  String? input = stdin.readLineSync();
  if(input != null) {
    age = int.parse(input);
    print('Next year you will be ${age + 1} years old');
  }
}
```

Alternatively, if you are rather certain that the input would rarely be null. You can force to get the not null data using “!”.

```
import 'dart:io';

void main() {
  int age = 0;
  stdout.write('Input age: ');
  String? input = stdin.readLineSync();
  age = int.parse(input!);
  print('Next year you will be ${age + 1} years old');
}
```

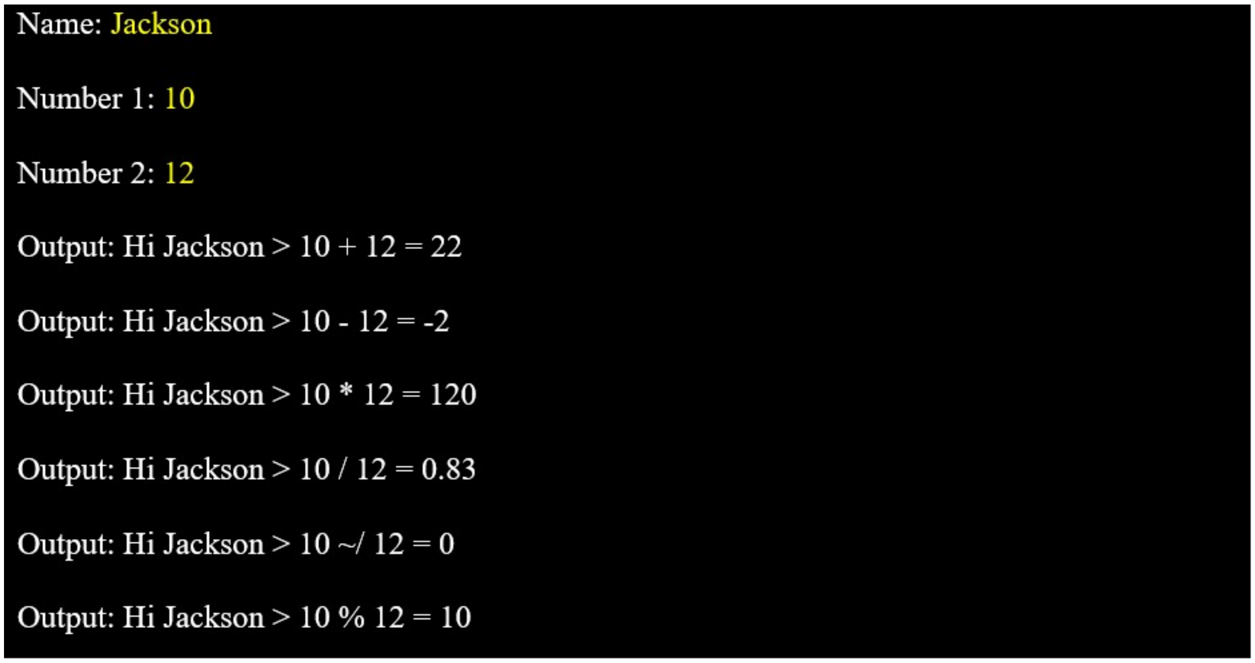
Some dart's method returns null if there is an error such as "tryParse()" method. It will help detect the error without using exception handling.

```
import 'dart:io';

void main() {
  stdout.write('Input age: ');
  String? input = stdin.readLineSync();
  // try to convert String to int
  int? age = int.tryParse(input!);
  if (age == null) {
    print('Please input only an integer!');
  } else {
    print('Next year you will be ${age + 1} years old');
  }
}
```

We can shorten the code above.

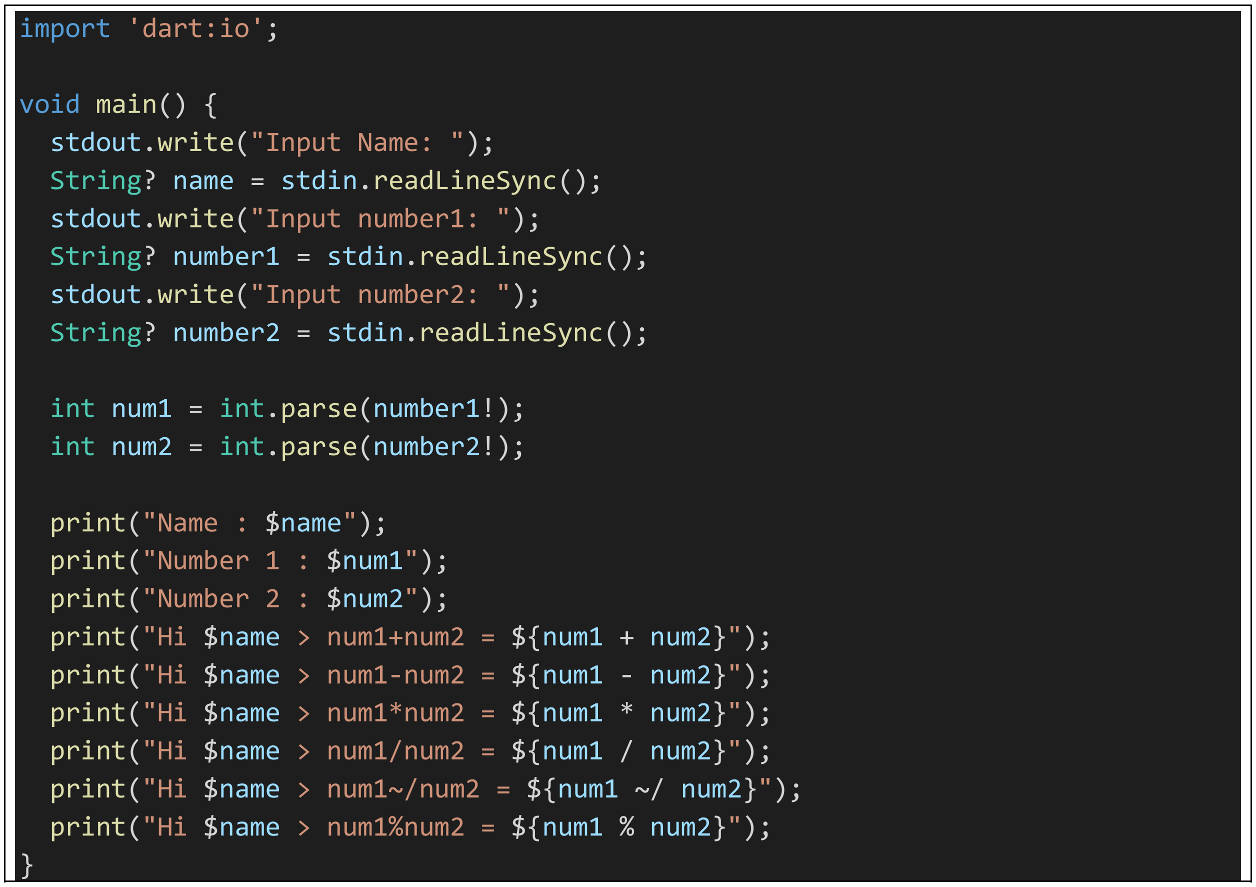
```
import 'dart:io';

void main() {
  stdout.write('Input age: ');
  // get String input and convert to int
  int? age = int.tryParse(stdin.readLineSync()!);
  // ternary operator
  age == null ? print('Please input only an integer!') : print('Next year you
will be ${age + 1} years old');
}
```

Exercise 1: (add, subtract, multiply, divide, divide and truncate, modulo) program



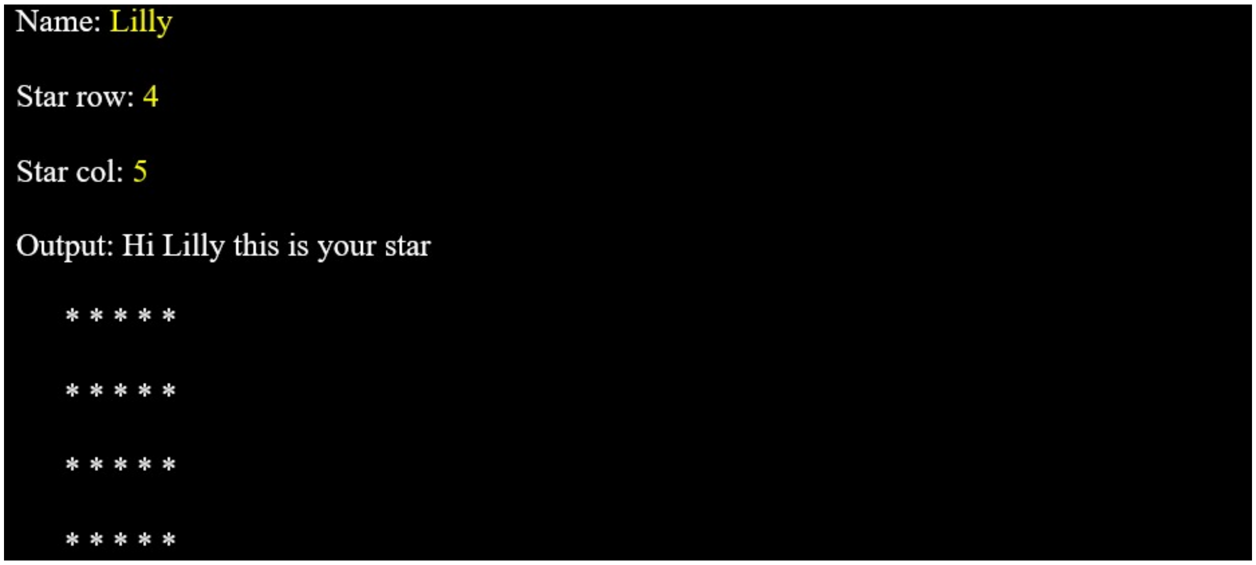
```
Name: Jackson

Number 1: 10

Number 2: 12

Output: Hi Jackson > 10 + 12 = 22

Output: Hi Jackson > 10 - 12 = -2

Output: Hi Jackson > 10 * 12 = 120

Output: Hi Jackson > 10 / 12 = 0.83

Output: Hi Jackson > 10 ~/ 12 = 0

Output: Hi Jackson > 10 % 12 = 10
```

<u>Your codes</u>



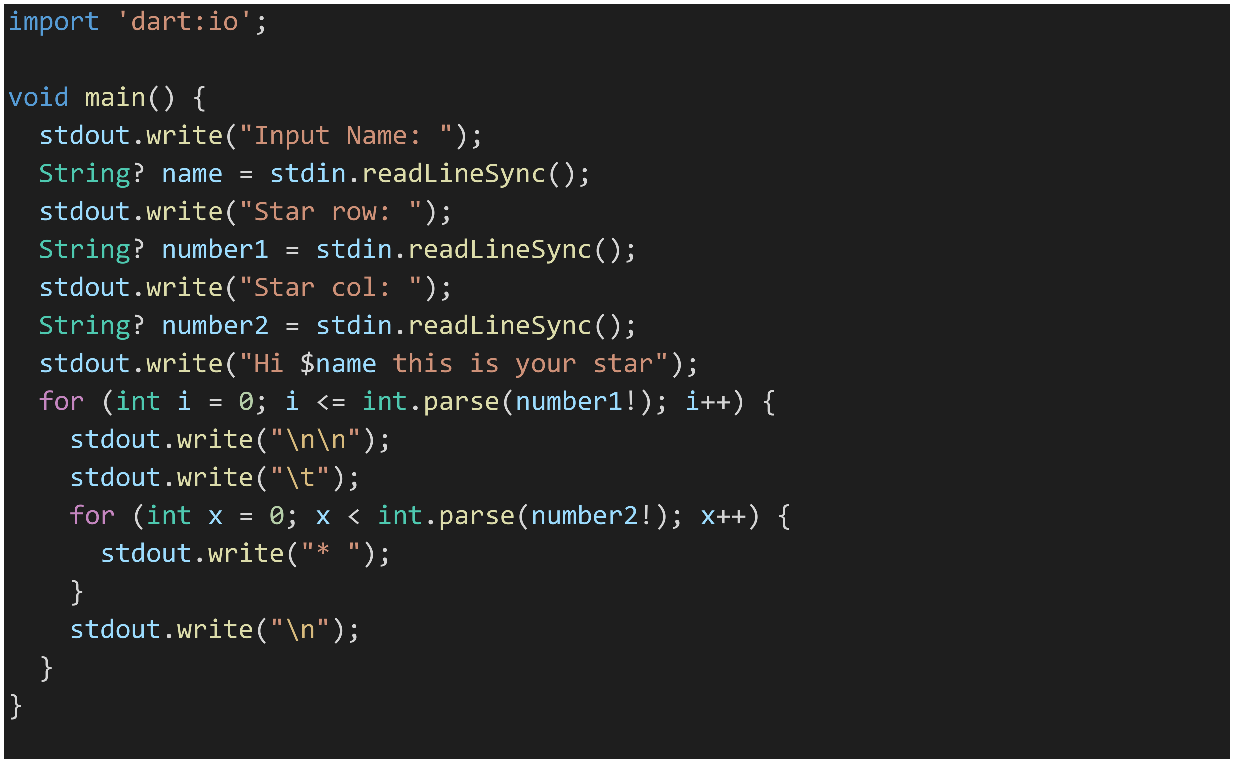
```dart
import 'dart:io';

void main() {
  stdout.write("Input Name: ");
  String? name = stdin.readLineSync();
  stdout.write("Input number1: ");
  String? number1 = stdin.readLineSync();
  stdout.write("Input number2: ");
  String? number2 = stdin.readLineSync();

  int num1 = int.parse(number1!);
  int num2 = int.parse(number2!);

  print("Name : $name");
  print("Number 1 : $num1");
  print("Number 2 : $num2");
  print("Hi $name > num1+num2 = ${num1 + num2}");
  print("Hi $name > num1-num2 = ${num1 - num2}");
  print("Hi $name > num1*num2 = ${num1 * num2}");
  print("Hi $name > num1/num2 = ${num1 / num2}");
  print("Hi $name > num1~/num2 = ${num1 ~/ num2}");
  print("Hi $name > num1%num2 = ${num1 % num2}");
}
```

Exercise 2: Star Program using loop



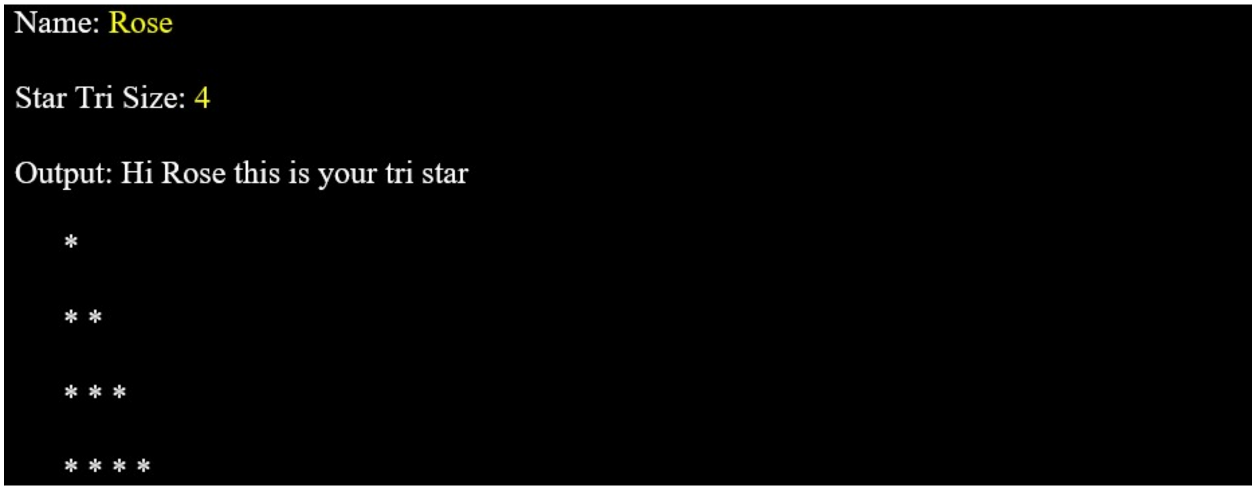
```
Name: Lilly

Star row: 4

Star col: 5

Output: Hi Lilly this is your star

    * * * * *

    * * * * *

    * * * * *

    * * * * *
```

Your codes



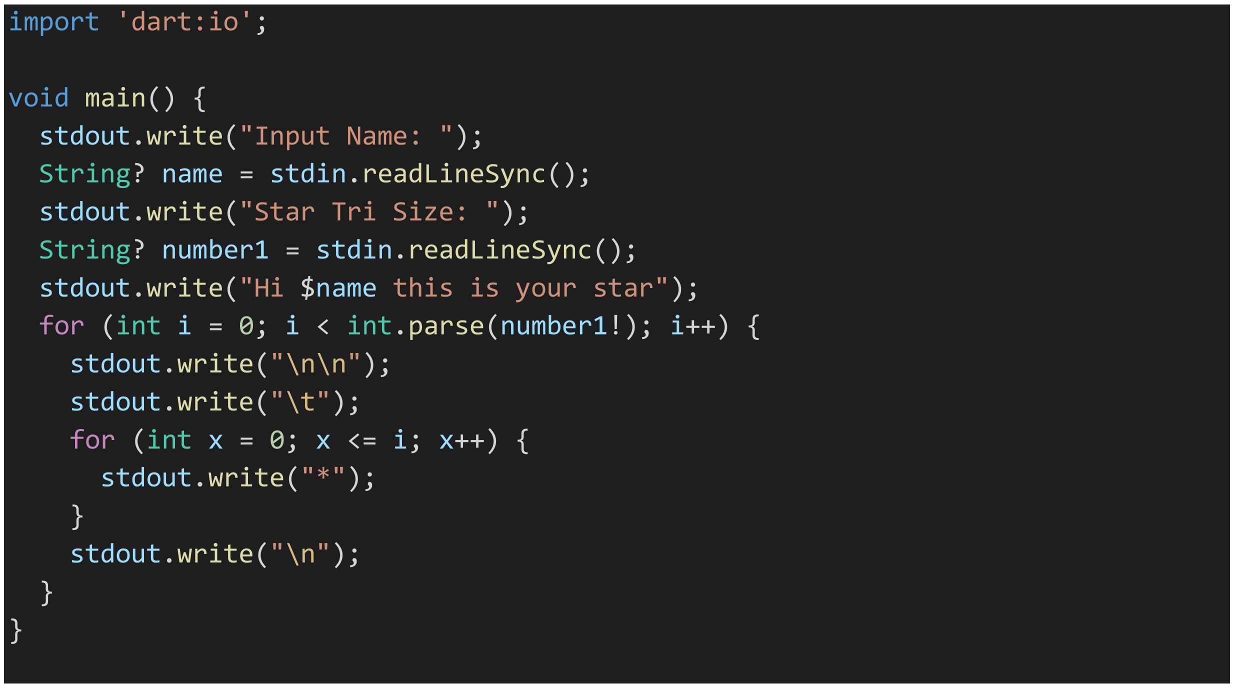
```
import 'dart:io';

void main() {
  stdout.write("Input Name: ");
  String? name = stdin.readLineSync();
  stdout.write("Star row: ");
  String? number1 = stdin.readLineSync();
  stdout.write("Star col: ");
  String? number2 = stdin.readLineSync();
  stdout.write("Hi $name this is your star");
  for (int i = 0; i <= int.parse(number1!); i++) {
    stdout.write("\n\n");
    stdout.write("\t");
    for (int x = 0; x < int.parse(number2!); x++) {
      stdout.write("* ");
    }
    stdout.write("\n");
  }
}
```

Exercise 3: Triangle star using loop



```
Name: Rose

Star Tri Size: 4

Output: Hi Rose this is your tri star

    *

    * *

    * * *

    * * * *
```

Your codes



```dart
import 'dart:io';

void main() {
  stdout.write("Input Name: ");
  String? name = stdin.readLineSync();
  stdout.write("Star Tri Size: ");
  String? number1 = stdin.readLineSync();
  stdout.write("Hi $name this is your star");
  for (int i = 0; i < int.parse(number1!); i++) {
    stdout.write("\n\n");
    stdout.write("\t");
    for (int x = 0; x <= i; x++) {
      stdout.write("*");
    }
    stdout.write("\n");
  }
}
```

Exercise 4: A decision and loop for weather display

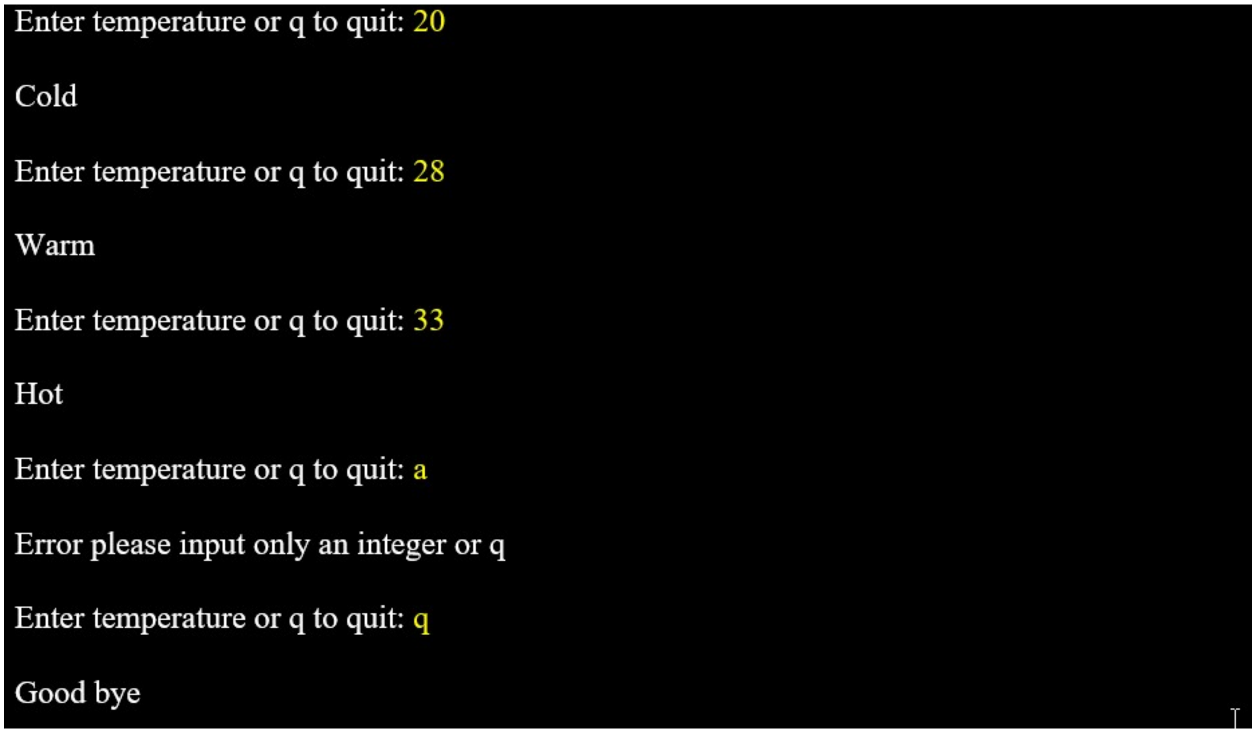
The program asks you to input a temperature in degree Celsius and display the result by these rules.

if temperature is lower than 25, show “Cold”.

if temperature is between 25 – 30, show “Warm”.

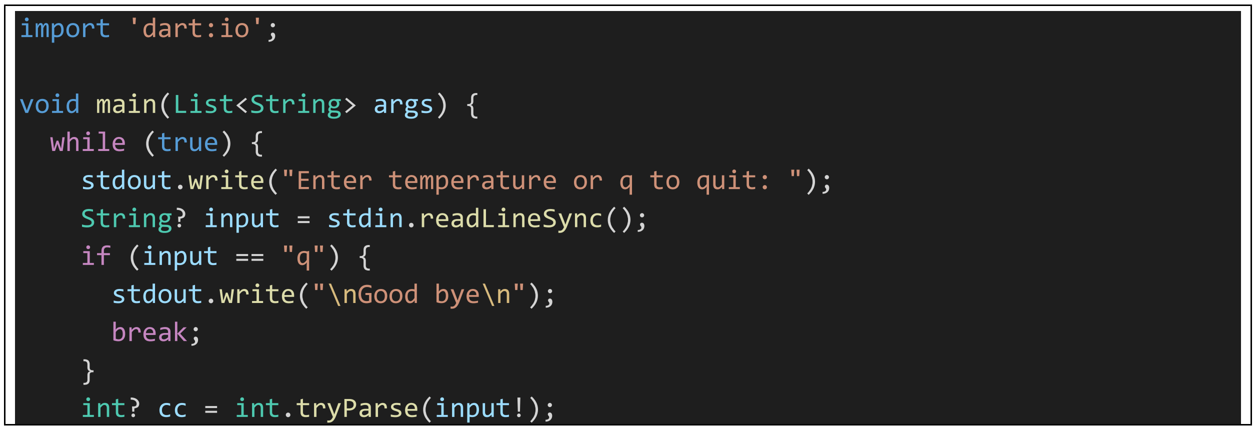
if temperature is more than 30, show “Hot”.

You can end the program by entering “q”.

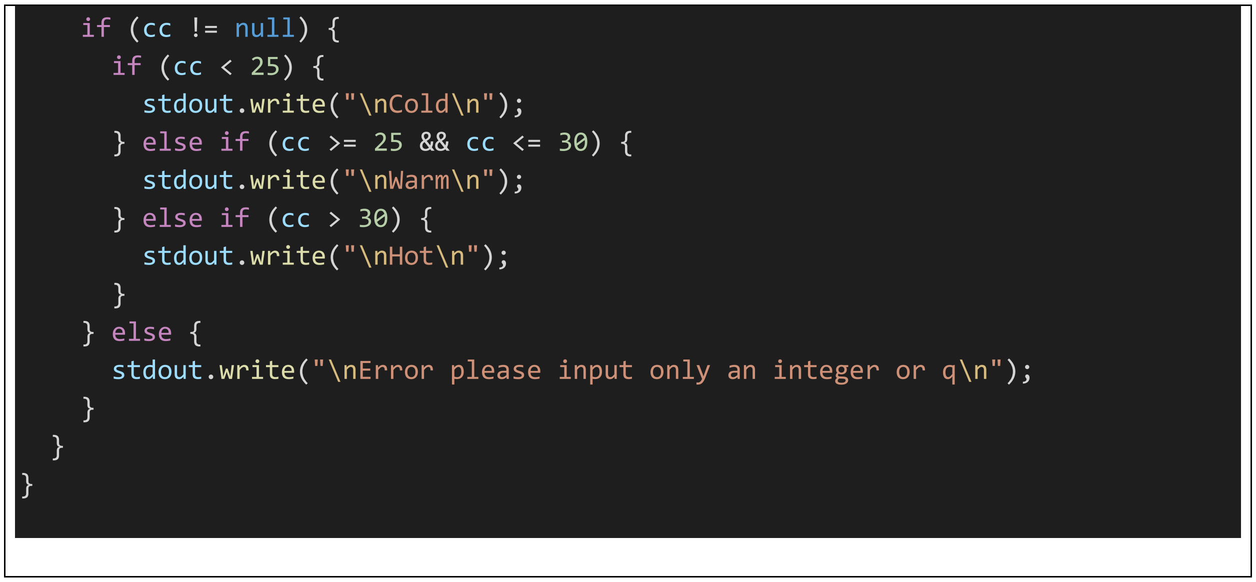


```
Enter temperature or q to quit: 20

Cold

Enter temperature or q to quit: 28

Warm

Enter temperature or q to quit: 33

Hot

Enter temperature or q to quit: a

Error please input only an integer or q

Enter temperature or q to quit: q

Good bye
```

Your codes

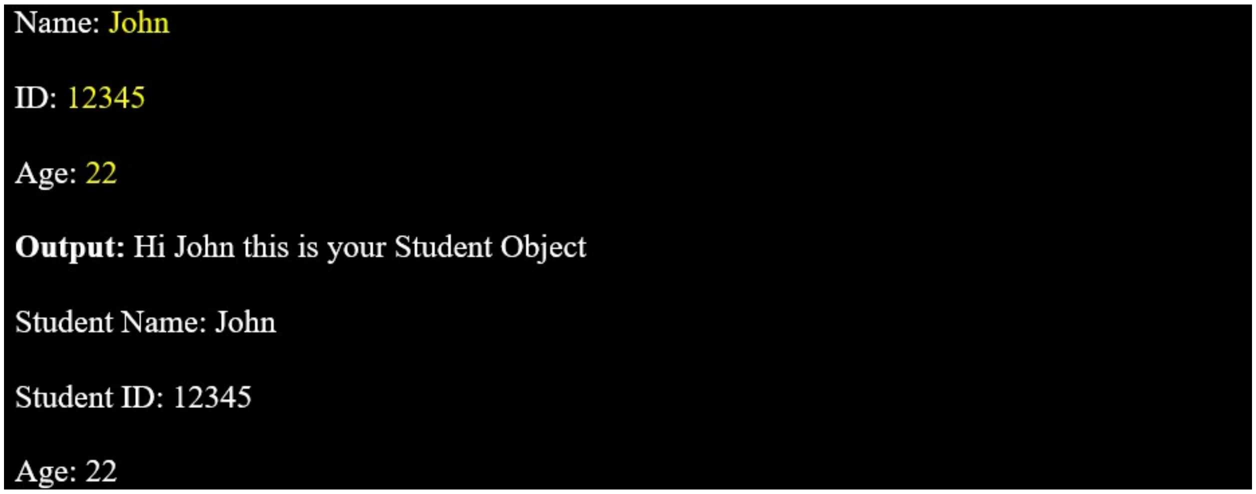


```dart
import 'dart:io';

void main(List<String> args) {
  while (true) {
    stdout.write("Enter temperature or q to quit: ");
    String? input = stdin.readLineSync();
    if (input == "q") {
      stdout.write("\nGood bye\n");
      break;
    }
    int? cc = int.tryParse(input!);
```



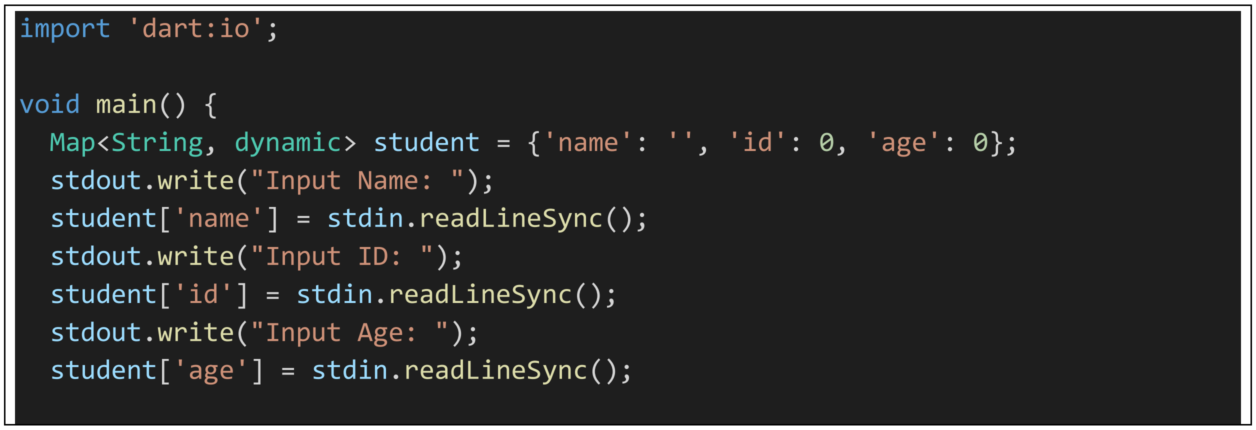
```dart
    if (cc != null) {
      if (cc < 25) {
        stdout.write("\nCold\n");
      } else if (cc >= 25 && cc <= 30) {
        stdout.write("\nWarm\n");
      } else if (cc > 30) {
        stdout.write("\nHot\n");
      }
    } else {
      stdout.write("\nError please input only an integer or q\n");
    }
  }
}
```

Exercise 5: Student Object



```
Name: John

ID: 12345

Age: 22

Output: Hi John this is your Student Object

Student Name: John

Student ID: 12345

Age: 22
```

Fill in the missing codes.

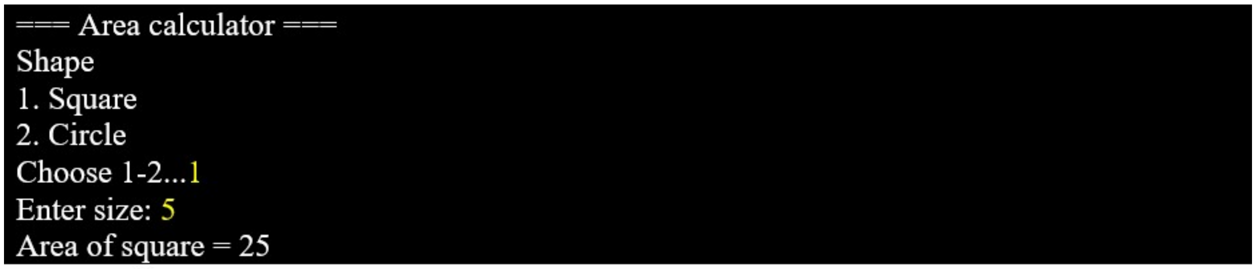


```dart
import 'dart:io';

void main() {
  Map<String, dynamic> student = {'name': '', 'id': 0, 'age': 0};
  stdout.write("Input Name: ");
  student['name'] = stdin.readLineSync();
  stdout.write("Input ID: ");
  student['id'] = stdin.readLineSync();
  stdout.write("Input Age: ");
  student['age'] = stdin.readLineSync();
  // output
  print('Output: Hi ${student['name']} this is your Student Object');
  print('Student Name: ${student['name']}');
  print('Student ID: ${student['id']}');
  print('Age: ${student['age']}');
}

import 'dart:io';

void main() {
  Map<String, dynamic> student = {'name': '', 'id': 0, 'age': 0};
  stdout.write("Input Name: ");
  student['name'] = stdin.readLineSync();
  stdout.write("Input ID: ");
  student['id'] = stdin.readLineSync();
  stdout.write("Input Age: ");
  student['age'] = stdin.readLineSync();

  // output
  print('Output: Hi ${student['name']} this is your Student Object');
  print('Student Name: ${student['name']}');
  print('Student ID: ${student['id']}');
  print('Age: ${student['age']}');
}
```

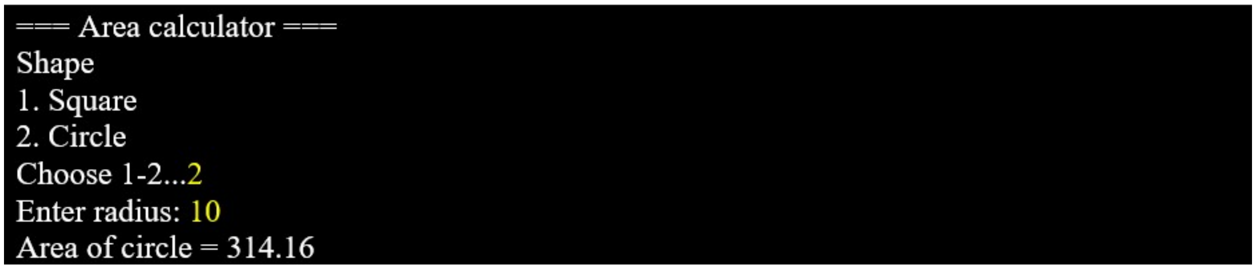
Exercise 6: Create two functions to calculate area of square and circle.

```
=== Area calculator ===
Shape
1. Square
2. Circle
Choose 1-2...x
Please input only integer
```

```
=== Area calculator ===
Shape
1. Square
2. Circle
Choose 1-2...1
Enter size: 5
Area of square = 25
```

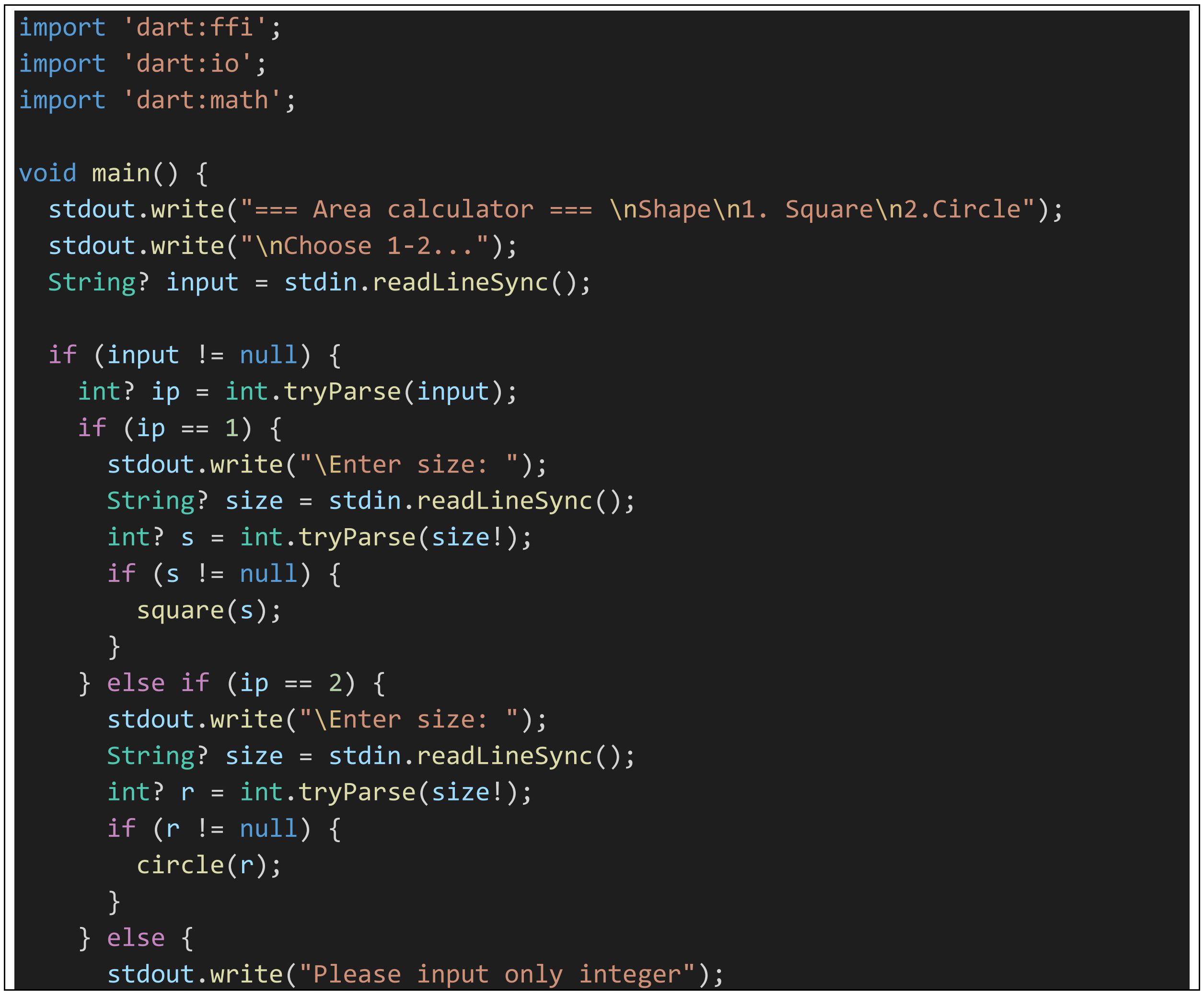


```
=== Area calculator ===
Shape
1. Square
2. Circle
Choose 1-2...1
Enter size: 5
Area of square = 25
```



```
=== Area calculator ===
Shape
1. Square
2. Circle
Choose 1-2...2
Enter radius: 10
Area of circle = 314.16
```

Fill in the missing codes.



```dart
import 'dart:ffi';
import 'dart:io';
import 'dart:math';

void main() {
  stdout.write("=== Area calculator === \nShape\n1. Square\n2.Circle");
  stdout.write("\nChoose 1-2...");
  String? input = stdin.readLineSync();

  if (input != null) {
    int? ip = int.tryParse(input);
    if (ip == 1) {
      stdout.write("\Enter size: ");
      String? size = stdin.readLineSync();
      int? s = int.tryParse(size!);
      if (s != null) {
        square(s);
      }
    } else if (ip == 2) {
      stdout.write("\Enter size: ");
      String? size = stdin.readLineSync();
      int? r = int.tryParse(size!);
      if (r != null) {
        circle(r);
      }
    } else {
      stdout.write("Please input only integer");
    }
  }
}

void square(int size) {
  stdout.write("Area if square = ${pow(size, size)}");
}

void circle(var r) {
  stdout.write("Area if square = ${pow(r, r) * pi}");
}
```

Assignment 1: A simple counter

A counter starts at 0. You can choose + to increase a counter by one or – to decrease a counter by one and choose 0 to end the program.

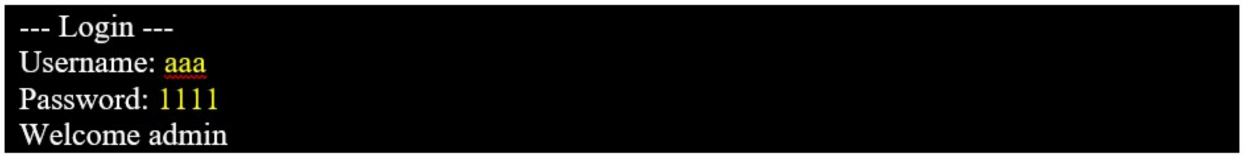
```
Count = 0
Increase (+) or decrease (-) or exit (0): +
Count = 1
Increase (+) or decrease (-) or exit (0): +
Count = 2
Increase (+) or decrease (-) or exit (0): -
Count = 1
Increase (+) or decrease (-) or exit (0): -
Count = 0
Increase (+) or decrease (-) or exit (0): -
Count = -1
Increase (+) or decrease (-) or exit (0): 0
Good bye
```

Your codes

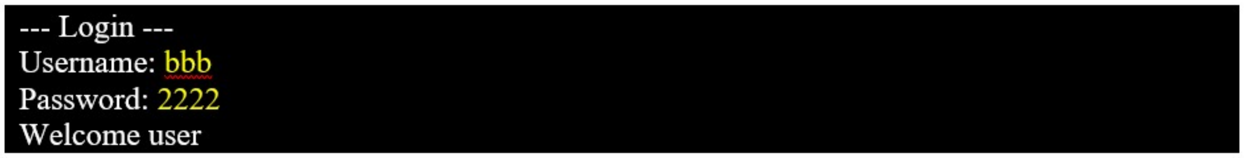
```dart
import 'dart:io';

void main(List<String> args) {
  int i = 0;
  while (true) {
    stdout.write('Count = $i \n');
    stdout.write('Increase(+) or decrease(-) or exit (0): ');
    String? input = stdin.readLineSync();
    if (input == '0') {
      break;
    } else if (input == '+') {
      i++;
    } else if (input == '-') {
      i--;
    } else {
      stdout.write('Error\n');
    }
  }
}
```

Assignment 2: A login function

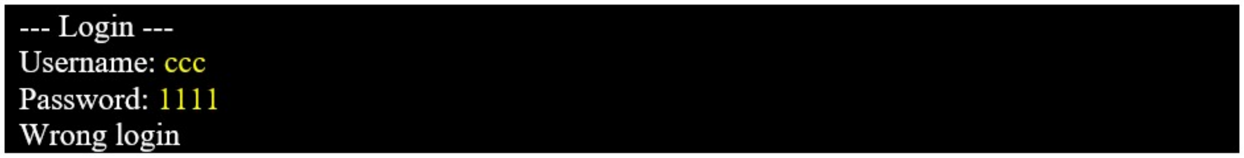
An app has fixed login accounts storing as a List of Map. Fill in missing codes to get the results below.



```
--- Login ---
Username: aaa
Password: 1111
Welcome admin
```

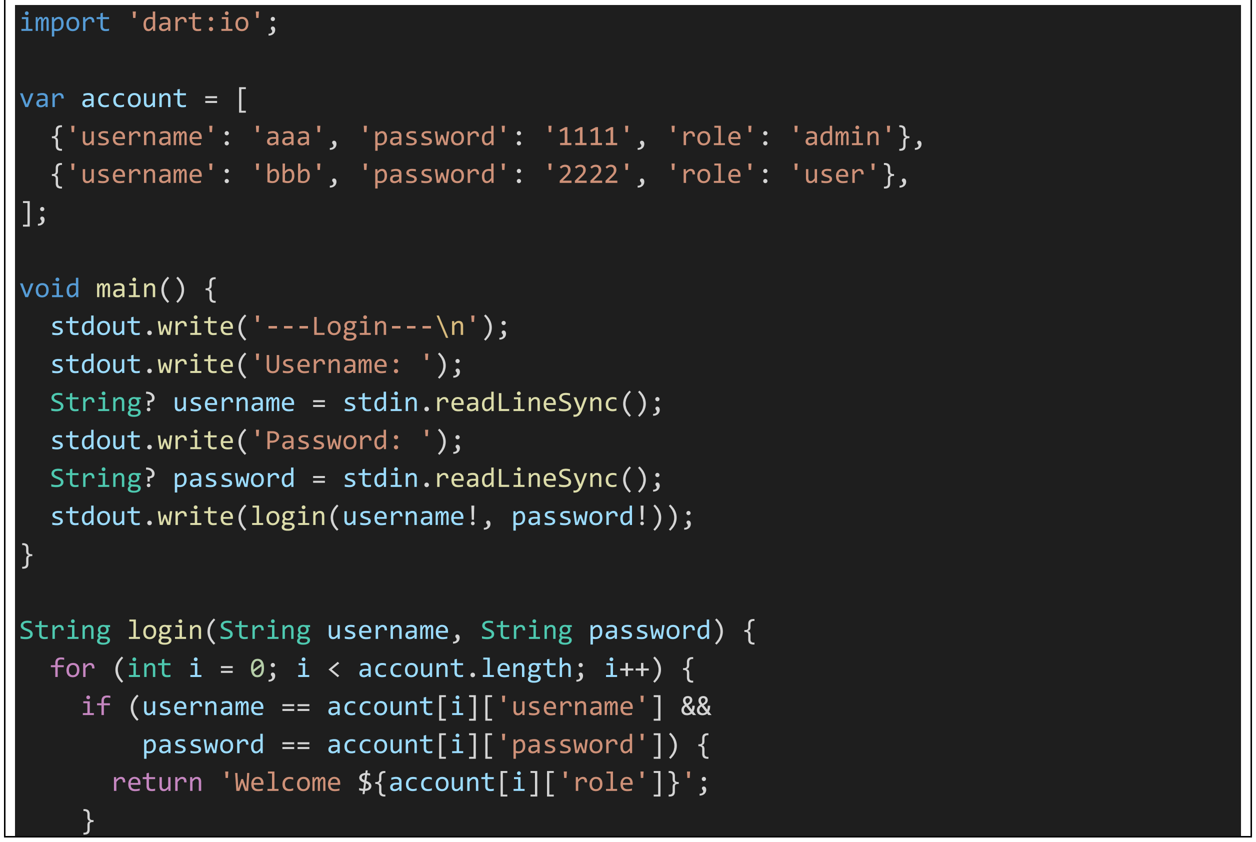


```
--- Login ---
Username: bbb
Password: 2222
Welcome user
```



```
--- Login ---
Username: ccc
Password: 1111
Wrong login
```

Your codes



```dart
import 'dart:io';

var account = [
  {'username': 'aaa', 'password': '1111', 'role': 'admin'},
  {'username': 'bbb', 'password': '2222', 'role': 'user'},
];

void main() {
  stdout.write('---Login---\n');
  stdout.write('Username: ');
  String? username = stdin.readLineSync();
  stdout.write('Password: ');
  String? password = stdin.readLineSync();
  stdout.write(login(username!, password!));
}

String login(String username, String password) {
  for (int i = 0; i < account.length; i++) {
    if (username == account[i]['username'] &&
        password == account[i]['password']) {
      return 'Welcome ${account[i]['role']}';
    }
  }
  return 'Wrong login';
}
```