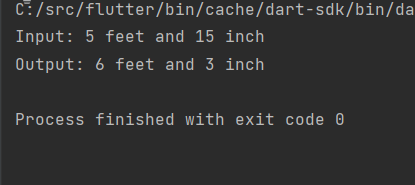
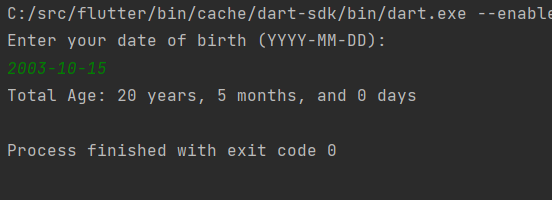
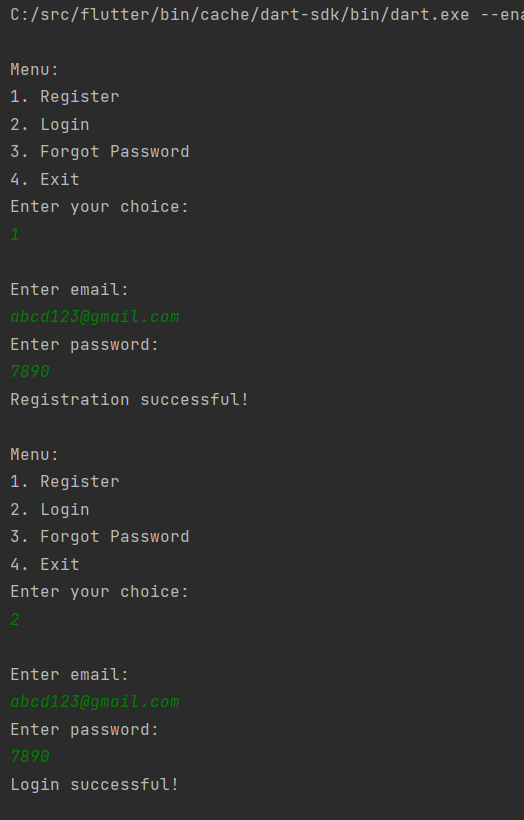
// ignore\_for\_file: avoid\_print  
  
class Distance{  
 int feet;  
 int inches;  
  
 Distance({required this.feet, required this.inches});  
  
 void convertToFeetAndInches() {  
 int totalInches = feet \* 12 + inches;  
 int convertedFeet = totalInches ~/ 12;  
 int convertedInches = totalInches % 12;  
  
 print('Output: $convertedFeet feet and $convertedInches inch');  
 }  
}  
  
void main() {  
 Distance converter = Distance(feet: 5, inches: 15);  
 print('Input: ${converter.feet} feet and ${converter.inches} inch');  
 converter.convertToFeetAndInches();  
}



// ignore\_for\_file: avoid\_print  
  
import 'dart:io';  
  
class AgeCalculator {  
 int birthYear;  
 int birthMonth;  
 int birthDay;  
  
 AgeCalculator({required this.birthYear, required this.birthMonth, required this.birthDay});  
  
 void calculateAge() {  
 DateTime currentDate = DateTime.now();  
 DateTime birthDate = DateTime(birthYear, birthMonth, birthDay);  
  
 int years = currentDate.year - birthDate.year;  
 int months = currentDate.month - birthDate.month;  
 int days = currentDate.day - birthDate.day;  
  
 if (months < 0 || (months == 0 && days < 0)) {  
 years--;  
 if (months < 0) {  
 months += 12;  
 }  
 }  
 if (days < 0) {  
 days += DateTime(birthDate.year + years, birthDate.month + months, 0).day;  
 months--;  
 }  
  
 print('Total Age: $years years, $months months, and $days days');  
 }  
}  
  
void main() {  
 print('Enter your date of birth (YYYY-MM-DD):');  
 String dobString = stdin.readLineSync()!;  
 List<String> dobParts = dobString.split('-');  
 int birthYear = int.*parse*(dobParts[0]);  
 int birthMonth = int.*parse*(dobParts[1]);  
 int birthDay = int.*parse*(dobParts[2]);  
  
 AgeCalculator calculator = AgeCalculator(birthYear: birthYear, birthMonth: birthMonth, birthDay: birthDay);  
 calculator.calculateAge();  
}



// ignore\_for\_file: avoid\_print  
  
import 'dart:io';  
  
class User {  
 String email;  
 String password;  
  
 User({required this.email, required this.password});  
}  
  
class LoginSystem {  
 List<Map<String, String>> users = [];  
  
 void register() {  
 print('\nEnter email:');  
 String email = stdin.readLineSync()!;  
 print('Enter password:');  
 String password = stdin.readLineSync()!;  
 users.add({'email': email, 'password': password});  
 print('Registration successful!');  
 }  
  
 bool login() {  
 print('\nEnter email:');  
 String email = stdin.readLineSync()!;  
 print('Enter password:');  
 String password = stdin.readLineSync()!;  
 for (var user in users) {  
 if (user['email'] == email && user['password'] == password) {  
 print('Login successful!');  
 return true;  
 }  
 }  
 print('Invalid email or password!');  
 return false;  
 }  
  
 void forgotPassword() {  
 print('\nEnter your email:');  
 String email = stdin.readLineSync()!;  
 for (var user in users) {  
 if (user['email'] == email) {  
 print('Enter new password:');  
 String newPassword = stdin.readLineSync()!;  
 user['password'] = newPassword;  
 print('Password updated successfully!');  
 return;  
 }  
 }  
 print('Email not found!');  
 }  
}  
  
void main() {  
 LoginSystem loginSystem = LoginSystem();  
  
 while (true) {  
 print('\nMenu:');  
 print('1. Register');  
 print('2. Login');  
 print('3. Forgot Password');  
 print('4. Exit');  
 print('Enter your choice:');  
  
 int choice = int.*parse*(stdin.readLineSync()!);  
  
 switch (choice) {  
 case 1:  
 loginSystem.register();  
 break;  
 case 2:  
 if (loginSystem.login()) {  
 return;  
 }  
 break;  
 case 3:  
 loginSystem.forgotPassword();  
 break;  
 case 4:  
 return;  
 default:  
 print('Invalid choice!');  
 }  
 }  
}



// ignore\_for\_file: avoid\_print  
  
class Person {  
 String name;  
 int age;  
 String location;  
  
 Person(this.name, this.age, this.location);  
  
 @override  
 String toString() {  
 return 'Name: $name, Age: $age, Location: $location';  
 }  
}  
  
void main() {  
 var person = Person('Jay', 20, 'Ahmedabad');  
  
 // Using cascade operator to perform multiple operations on the object  
 var updatedPerson = Person('priya',25,'Gandhinagar')  
 ..name = 'Vishakha'  
 ..age = person.age  
 ..location = 'Ahmedabad';  
  
 print('Original Person: $person');  
 print('Updated Person: $updatedPerson');  
}

