Solution

1.

```
void main(){
List<int> a = [1,1,2,3,5,8,13,21,34,55,89];
for(int element in a){
if(element<5){
print(element);
}
}
}
```

2.

```
List<int> findCommonElements(List<int> list1, List<int> list2) {
  Set<int> set1 = list1.toSet();
  Set<int> set2 = list2.toSet();
Set<int> commonElements = set1.intersection(set2);
 return commonElements.toList();
}
void main() {
  List<int> a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89];
  List<int> b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13];
  List<int> commonElements = findCommonElements(a, b);
  print(commonElements);
}
```

3.

```
bool isPalindrome(String input) {
  int start = 0;
  int end = input.length - 1;
  while (start < end) {
```

```
    if (input[start] != input[end]) {
      return false;
    }
    start++;
    end--;
  }
 return true;
}
void main() {
String testString1 = "mum";
String testString2 = "hello";
String testString3 = "racecar";
print("$testString1 is a palindrome: ${isPalindrome(testString1)}");
print("$testString2 is a palindrome: ${isPalindrome(testString2)}");
print("$testString3 is a palindrome: ${isPalindrome(testString3)}");
}
```

4.

```
List<int> getFirstAndLast(List<int> inputList) {
  if (inputList.isEmpty) {
    return [];
  }
 return [inputList.first, inputList.last];
}
void main() {
  List<int> a = [5, 10, 15, 20, 25];
  List<int> result = getFirstAndLast(a);
  print(result);
}
```

5.

```
import 'dart:io';

String reverseWords(String input) {
 List<String> words = input.split(' ');
 words = words.reversed.toList();
 return words.join(' ');
}
void main() {
print("Enter a long string containing multiple words:");
String userInput = stdin.readLineSync() ?? "";
String reversedString = reverseWords(userInput);
print("Reversed Order: $reversedString");
}
```

6.

```
List<T> removeDuplicates<T>(List<T> inputList) {
Set<T> uniqueElements = inputList.toSet();
return uniqueElements.toList();
}
void main() {
List<int> inputList = [1, 2, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, 9];
List<int> result = removeDuplicates(inputList);
print(result);
// Output: [1, 2, 3, 4, 5, 6, 7, 8, 9]
}
```

7.

No file given