**ArrayList Exercises:**

**Exercise 1: Creating and Modifying ArrayList**

**Create an ArrayList of integers and perform the following operations:**

**Add the numbers 1 to 5 to the ArrayList.**

**Insert the number 6 at the beginning of the ArrayList.**

**Add the number 7 at the end of the ArrayList.**

**Replace the number at index 3 with 8.**

**Remove the number 2 from the ArrayList.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<Integer> numbers = new ArrayList<>();

// Add numbers 1 to 5

for (int i = 1; i <= 5; i++) {

numbers.add(i);

}

// Insert 6 at the beginning

numbers.add(0, 6);

// Add 7 at the end

numbers.add(7);

// Replace element at index 3 with 8

numbers.set(3, 8);

// Remove number 2

numbers.remove(Integer.valueOf(2));

System.out.println(numbers);

}

}

**Exercise 2: Iterating through an ArrayList**

**Create an ArrayList of strings and iterate through it using a for-each loop to print each element.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> fruits = new ArrayList<>();

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Cherry");

for (String fruit : fruits) {

System.out.println(fruit);

}

}

}

**Exercise 3: Searching and Removing Elements**

**Create an ArrayList of names and perform the following operations:**

**Check if "John" is in the list and print whether it's found.**

**Remove "Alice" from the list if it exists.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> names = new ArrayList<>();

names.add("John");

names.add("Alice");

names.add("Bob");

// Check if "John" is in the list

boolean containsJohn = names.contains("John");

System.out.println("Contains John: " + containsJohn);

// Remove "Alice" if it exists

if (names.contains("Alice")) {

names.remove("Alice");

}

}

}

**Exercise 1:**

**Create an ArrayList and add elements to it**.

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

System.out.println(arrayList);

}

}

**Exercise 2:**

**Access an element at a specific index in an ArrayList.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

String fruit = arrayList.get(1);

System.out.println(fruit);

}

}

**Exercise 3:**

**Remove an element from an ArrayList using index.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

arrayList.remove(1);

System.out.println(arrayList);

}

}

**Exercise 4:**

**Check if an element exists in an ArrayList.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

boolean containsBanana = arrayList.contains("Banana");

System.out.println(containsBanana);

}

}

**Exercise 5:**

**Find the size of an ArrayList**.

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

int size = arrayList.size();

System.out.println(size);

}

}

**Exercise 6:**

**Replace an element at a specific index in an ArrayList.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

arrayList.set(1, "Mango");

System.out.println(arrayList);

}

}

**Exercise 7:**

**Remove all elements from an ArrayList.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

arrayList.clear();

System.out.println(arrayList);

}

}

**Exercise 8:**

**Loop through an ArrayList using a for-each loop.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

for (String fruit : arrayList) {

System.out.println(fruit);

}

}

}

**LinkedList Exercises:**

**Exercise 9:**

**Create a LinkedList and add elements to it.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

System.out.println(linkedList);

}

}

**Exercise 10:**

**Access an element at a specific index in a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

String animal = linkedList.get(1);

System.out.println(animal);

}

}

**Exercise 11:**

**Remove an element from a LinkedList using index.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

linkedList.remove(1);

System.out.println(linkedList);

}

}

**Exercise 12:**

**Check if an element exists in a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

boolean containsDog = linkedList.contains("Dog");

System.out.println(containsDog);

}

}

**Exercise 13:**

**Find the size of a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

int size = linkedList.size();

System.out.println(size);

}

}

**Exercise 14:**

**Replace an element at a specific index in a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

linkedList.set(1, "Tiger");

System.out.println(linkedList);

}

}

**Exercise 15:**

**Remove all elements from a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

linkedList.clear();

System.out.println(linkedList);

}

}

**Exercise 16:**

**Loop through a LinkedList using an iterator.**

import java.util.LinkedList;

import java.util.Iterator;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

Iterator<String> iterator = linkedList.iterator();

while (iterator.hasNext()) {

System.out.println(iterator.next());

}

}

}

**ArrayList Exercises:**

**Exercise 17:**

**Write a program to find the index of the first occurrence of a specific element in an ArrayList.**

import java.util.ArrayList;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

int index = arrayList.indexOf("Banana");

System.out.println(index);

}

}

**Exercise 18:**

**Write a program to sort an ArrayList of integers in ascending order.**

import java.util.ArrayList;

import java.util.Collections;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<Integer> arrayList = new ArrayList<>();

arrayList.add(3);

arrayList.add(1);

arrayList.add(2);

Collections.sort(arrayList);

System.out.println(arrayList);

}

}

**LinkedList Exercises:**

**Exercise 19:**

**Write a program to add an element at the beginning of a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Dog");

linkedList.add("Elephant");

linkedList.addFirst("Cat");

System.out.println(linkedList);

}

}

**Exercise 20:**

**Write a program to reverse the order of elements in a LinkedList.**

import java.util.LinkedList;

import java.util.Collections;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

Collections.reverse(linkedList);

System.out.println(linkedList);

}

}

**Exercise 21:**

**Write a program to find and remove the last element from a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Cat");

linkedList.add("Dog");

linkedList.add("Elephant");

linkedList.removeLast();

System.out.println(linkedList);

}

}

**Exercise 22:**

**Write a program to clone a LinkedList into another LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> originalList = new LinkedList<>();

originalList.add("Apple");

originalList.add("Banana");

originalList.add("Orange");

LinkedList<String> clonedList = new LinkedList<>(originalList);

System.out.println(clonedList);

}

}

**Exercise 23:**

**Write a program to find the first and last occurrence of a specific element in a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

int firstIndex = linkedList.indexOf("Banana");

int lastIndex = linkedList.lastIndexOf("Banana");

System.out.println("First Index: " + firstIndex);

System.out.println("Last Index: " + lastIndex);

}

}

**Exercise 24:**

**Write a program to shuffle the elements of a LinkedList.**

import java.util.LinkedList;

import java.util.Collections;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

Collections.shuffle(linkedList);

System.out.println(linkedList);

}

}

**Exercise 25:**

**Write a program to check if a LinkedList is empty.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

System.out.println("Is the LinkedList empty? " + linkedList.isEmpty());

linkedList.add("Cat");

linkedList.add("Dog");

System.out.println("Is the LinkedList empty now? " + linkedList.isEmpty());

}

}

ArrayList Exercises:

**Exercise 26:**

**Write a program to remove duplicates from an ArrayList.**

import java.util.ArrayList;

import java.util.HashSet;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

arrayList.add("Apple");

HashSet<String> uniqueSet = new HashSet<>(arrayList);

arrayList.clear();

arrayList.addAll(uniqueSet);

System.out.println(arrayList);

}

}

**Exercise 27:**

**Write a program to find the largest element in an ArrayList of integers.**

import java.util.ArrayList;

import java.util.Collections;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<Integer> arrayList = new ArrayList<>();

arrayList.add(25);

arrayList.add(17);

arrayList.add(32);

Integer max = Collections.max(arrayList);

System.out.println("Largest element: " + max);

}

}

**LinkedList Exercises:**

**Exercise 28:**

**Write a program to add elements from another collection at the end of a LinkedList.**

import java.util.LinkedList;

import java.util.Arrays;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.addAll(Arrays.asList("Orange", "Mango"));

System.out.println(linkedList);

}

}

**Exercise 29:**

**Write a program to find the sum of all elements in a LinkedList of integers.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<Integer> linkedList = new LinkedList<>();

linkedList.add(5);

linkedList.add(10);

linkedList.add(15);

int sum = 0;

for (Integer num : linkedList) {

sum += num;

}

System.out.println("Sum: " + sum);

}

}

**Exercise 30:**

**Write a program to sort a LinkedList of strings in alphabetical order.**

import java.util.LinkedList;

import java.util.Collections;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Orange");

linkedList.add("Apple");

linkedList.add("Banana");

Collections.sort(linkedList);

System.out.println(linkedList);

}

}

**Exercise 31:**

**Write a program to concatenate two LinkedLists.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> list1 = new LinkedList<>();

list1.add("Apple");

list1.add("Banana");

LinkedList<String> list2 = new LinkedList<>();

list2.add("Orange");

list2.add("Mango");

list1.addAll(list2);

System.out.println(list1);

}

}

**Exercise 32:**

**Write a program to remove all occurrences of a specific element from a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Apple");

linkedList.add("Orange");

linkedList.add("Apple");

linkedList.removeAll(Collections.singleton("Apple"));

System.out.println(linkedList);

}

}

**Exercise 33:**

**Write a program to get a sublist from a LinkedList.**

import java.util.LinkedList;

import java.util.List;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

linkedList.add("Mango");

List<String> sublist = linkedList.subList(1, 3);

System.out.println(sublist);

}

}

**Exercise 34:**

**Write a program to reverse the elements of a LinkedList.**

import java.util.LinkedList;

import java.util.Collections;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

Collections.reverse(linkedList);

System.out.println(linkedList);

}

}

**Exercise 35:**

**Write a program to check if two LinkedLists are equal.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> list1 = new LinkedList<>();

list1.add("Apple");

list1.add("Banana");

LinkedList<String> list2 = new LinkedList<>();

list2.add("Apple");

list2.add("Banana");

boolean isEqual = list1.equals(list2);

System.out.println("Are the LinkedLists equal? " + isEqual);

}

}

**ArrayList Exercises:**

**Exercise 36:**

**Write a program to find the frequency of a specific element in an ArrayList.**

import java.util.ArrayList;

import java.util.Collections;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<Integer> arrayList = new ArrayList<>();

arrayList.add(5);

arrayList.add(2);

arrayList.add(8);

arrayList.add(5);

arrayList.add(3);

int frequency = Collections.frequency(arrayList, 5);

System.out.println("Frequency of 5: " + frequency);

}

}

**Exercise 37:**

**Write a program to copy the elements of one ArrayList to another.**

import java.util.ArrayList;

import java.util.Collections;

public class ArrayListExercise {

public static void main(String[] args) {

ArrayList<String> sourceList = new ArrayList<>();

sourceList.add("Apple");

sourceList.add("Banana");

ArrayList<String> targetList = new ArrayList<>(sourceList);

System.out.println(targetList);

}

}

**LinkedList Exercises:**

**Exercise 38:**

**Write a program to find the middle element of a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<Integer> linkedList = new LinkedList<>();

linkedList.add(1);

linkedList.add(2);

linkedList.add(3);

linkedList.add(4);

int size = linkedList.size();

int middleIndex = size / 2;

int middleElement = linkedList.get(middleIndex);

System.out.println("Middle Element: " + middleElement);

}

}

**Exercise 39:**

**Write a program to remove the first and last elements from a LinkedList**.

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

linkedList.removeFirst();

linkedList.removeLast();

System.out.println(linkedList);

}

}

**Exercise 40:**

**Write a program to find the maximum and minimum elements in a LinkedList.**

import java.util.LinkedList;

import java.util.Collections;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<Integer> linkedList = new LinkedList<>();

linkedList.add(5);

linkedList.add(10);

linkedList.add(3);

linkedList.add(8);

Integer max = Collections.max(linkedList);

Integer min = Collections.min(linkedList);

System.out.println("Max: " + max);

System.out.println("Min: " + min);

}

}

**Exercise 41:**

**Write a program to convert a LinkedList to an array.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

String[] array = linkedList.toArray(new String[0]);

for (String fruit : array) {

System.out.println(fruit);

}

}

}

**Exercise 42:**

**Write a program to find the occurrence of a specific element in a LinkedList.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Apple");

linkedList.add("Orange");

int occurrence = 0;

for (String fruit : linkedList) {

if (fruit.equals("Apple")) {

occurrence++;

}

}

System.out.println("Occurrence of Apple: " + occurrence);

}

}

**Exercise 43:**

**Write a program to swap two elements in a LinkedList.**

import java.util.LinkedList;

import java.util.Collections;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<Integer> linkedList = new LinkedList<>();

linkedList.add(5);

linkedList.add(10);

linkedList.add(3);

Collections.swap(linkedList, 0, 2);

System.out.println(linkedList);

}

}

**Exercise 44:**

**Write a program to clear all elements from a LinkedList except the first and last elements.**

import java.util.LinkedList;

public class LinkedListExercise {

public static void main(String[] args) {

LinkedList<String> linkedList = new LinkedList<>();

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Orange");

linkedList.add("Mango");

while (linkedList.size() > 2) {

linkedList.remove(1);

}

System.out.println(linkedList);

}

}