**Q1.**

package week\_4;

import java.util.ArrayList;

import java.util.List;

/\*\*

\* ArrayList practice.

\*

\* Remove "Oatmeal" from the ArrayList.

\* Add the name of your favorite breakfast food to the ArrayList.

\* Add "Cornflakes" to the ArrayList.

\* Display all of the items in the ArrayList.

\* Print a message if the ArrayList contains “Special K”.

\* Print a different message if it does not contain "Special K".

\*

\* (optional) non-programming question: what does Captain Crunch have to do with computer hacking?

\*

\*/

public class Question\_1\_Breakfast {

public static void main(String[] args) {

breakfast();

}

public static List<String> breakfast() {

// Creating a new ArrayList.

List<String> cereals = new ArrayList<>();

// Don't modify these three lines

cereals.add("Special K");

cereals.add("Captain Crunch");

cereals.add("Oatmeal");

// Removing "Oatmeal" from the ArrayList.

cereals.remove("Oatmeal");

// Adding the name of your favorite breakfast food to the List.

cereals.add("Pancakes");

// Adding the String "Cornflakes" to the List.

cereals.add("Cornflakes");

// Printing all of the items in the ArrayList, one per line. Use a loop.

for (String cereal : cereals) {

System.out.println(cereal);

}

// Checking if the ArrayList contains "Special K".

if (cereals.contains("Special K")) {

System.out.println("Special K is in the list.");

} else {

System.out.println("Special K is not in the list.");

}

// Printing a message with the number of items in the list.

System.out.println("The number of items in the list is: " + cereals.size());

// (optional) non-programming question: what does Captain Crunch have to do with computer hacking?

// This line needs to be the last line in this method,

// so write all of your code before this line.

// Don't modify this line.

// The test needs the method to return your ArrayList after the modifications you make.

return cereals;

}

}

**Q2.**

package week\_4;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

import static input.InputUtils.positiveIntInput;

import static input.InputUtils.yesNoInput;

/\*\*

\* Finish this program to roll a set of dice. Generate a random number between 1 and 6 for

\* each dice to be rolled, and save the values in a list.

\*

\* Display the total of all the dice rolled.

\*

\* In some games, rolling the same number on all dice has a special meaning.

\* In your program, check if all dice have the same value, and print a message

\* if all the dice show the same value. In other words, you'll need to write a method that

\* checks if all of the values in a list are the same.

\*

\*/

public class Question\_2\_Dice\_Roll {

public static final String SAME\_VALUES = "All the dice have the same value!";

static Random rnd = new Random(); // You will use this Random number generator in your roll method.

public static void main(String[] args) {

// How many dice to roll?

int numberOfDice = positiveIntInput("How many dice would you like to roll?");

// A do loop is similar to a while loop, but the condition is

// checked at the end of one loop iteration.

// So the set of dice are always rolled at least one time.

do {

// Roll the dice

List<Integer> diceValues = roll(numberOfDice);

// Print the dice values rolled

System.out.println("The dice have the values: " + diceValues);

System.out.println("The total of all dice: " + diceTotal(diceValues));

if (allSameValue(diceValues)) {

System.out.println(SAME\_VALUES);

}

} while (yesNoInput("Do you want to roll again?"));

}

/\*\*

\* This method rolls a given number of dice.

\*

\* @param numberOfDice number of dice to roll.

\* @return a list containing the values of each rolled dice.

\* If the numberOfDice is 0 or negative, return an empty list.

\*/

public static List<Integer> roll(int numberOfDice) {

// Use the Random rnd variable declared on line 28 to generate random numbers.

// Don't create another Random object.

// Create an ArrayList of Integer values.

List<Integer> diceValues = new ArrayList<>();

// Use a loop to roll the given number of dice. Store the values in an ArrayList and return it.

for (int i = 0; i < numberOfDice; i++) {

diceValues.add(rnd.nextInt(6) + 1);

}

// If the numberOfDice is 0 or negative, return an empty List.

if (numberOfDice <= 0) {

return new ArrayList<>();

}

return diceValues;

}

/\*\*

\* This method calculates the total value of all dice rolled.

\*

\* @param diceValues a list containing the values of each rolled dice.

\* @return the total value of all dice rolled.

\* If the diceValues List is null or empty, return 0.

\*/

public static int diceTotal(List<Integer> diceValues) {

if (diceValues == null || diceValues.isEmpty()) {

return 0; }

int total = 0;

for (int value : diceValues) {

total += value;

}

return total;

}

public static boolean allSameValue(List<Integer> diceValues) {

if (diceValues == null || diceValues.isEmpty()) {

return false;

}

int firstValue = diceValues.get(0);

for (int i = 1; i < diceValues.size(); i++) {

if (diceValues.get(i) != firstValue) {

return false;

}

}

return true;

}

}

**Q3.**

package week\_4;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import static input.InputUtils.stringInput;

import static input.InputUtils.yesNoInput;

/\*\*

Finish the methods in this program that creates and manages a movie watchlist.

Follow the instructions in the methods.

\*/

public class Question\_3\_Movie\_Watch\_List {

public static void main(String[] args) {

// Create new ArrayList to contain names of movies to watch

List<String> movieWatchList = new ArrayList<>();

// Adding movies

do {

String newMovieName = stringInput("Enter name of movie to watch");

addMovie(newMovieName, movieWatchList);

} while (yesNoInput("Add more movies?"));

// Get the next movie to watch

String nextMovie = getNextMovie(movieWatchList);

if (nextMovie != null) {

System.out.println("The next movie to watch is " + nextMovie);

} else {

System.out.println("There are no movies in the watchlist.");

}

// Getting a random movie

String randomMovie = getRandomMovieFromWatchList(movieWatchList);

if (randomMovie != null) {

System.out.println("A random movie from the list is " + randomMovie);

} else {

System.out.println("There are no movies in the watchlist.");

}

// Removing a movie - you've watched it, or don't want to watch it

String movieToRemove = stringInput("Enter the name of a movie to remove");

removeMovie(movieToRemove, movieWatchList);

// Display all movies in order added

System.out.println("\n \*\* All movies, in order to watch \*\*\n");

printMoviesInWatchListOrder(movieWatchList);

// Display all movies in name order

System.out.println("\n \*\* All movies, in name order \*\*\n");

printMoviesInNameOrder(movieWatchList);

}

public static void addMovie(String movie, List<String> movies) {

/\* Part 1.

TODO finish this method.

This method should add the String movie to the END of the movies List,

but only if the movie is not in the list.

Don't change the case of the movie string when adding it to the movies list.

If the movie is 'WALL-E' then add this exact string.

If the movie is 'Star Wars: Episode IV – A New Hope' add this exact string.

If the movies list contains ['Up', 'Jaws', 'Spiderman']

and the movie String is 'Rocky' then it should be added to the end of the list.

The movies list will become ['Up', 'Jaws', 'Spiderman', 'Rocky']

Print the message "Movie added!"

Don't add the movie if it is already in the movies list.

Your check should be case-insensitive.

If the movies list contains ['Up', 'Jaws', 'Spiderman']

and if the movie String is 'Up' then it should NOT be added.

or, if the movie String is 'up' then it should NOT be added.

or, if the movie String is 'UP' then it should NOT be added.

If the movie is already in the list, print the message "This movie is already in your watchlist!"

This method does not need to return anything.

\*/

// convert movie string to lowercase for case-insensitive comparison

String lowercaseMovie = movie.toLowerCase();

// check if movie is already in the list

if (movies.contains(lowercaseMovie)) {

System.out.println("This movie is already in your watchlist!");

} else {

// add movie to the end of the list

movies.add(movie);

System.out.println("Movie added!");

}

}

public static String getNextMovie(List<String> movies) {

/\* Part 2.

TODO finish this method.

If the movies list is not null, and has as at least one movie in it,

return the first movie in the list.

Don't modify the movies list.

If the movies list is null, or empty, return null.

Hint: check if the list is null or empty first.

\*/

// Check if the list is null or empty

if (movies == null || movies.isEmpty()) {

return null;

}

// Return the first movie in the list

return movies.get(0);

}

public static void removeMovie(String movie, List<String> movies) {

/\* Part 3.

TODO finish this method.

Remove the movie from the movies list.

Your check should be case-insensitive.

If the movie is in the movies list, remove that movie and print the

message "Movie removed!"

If the movies list contains ['Up', 'Jaws', 'Spiderman']

and the movie String is 'Jaws' then the 'Jaws' entry in the list should be removed.

or if the movie String is 'jaws' then the 'Jaws' entry in the list should be removed.

or if the movie String is 'JAWS' then the 'Jaws' entry in the list should be removed.

Print the message "Movie removed!"

If the movies list contains ['Up', 'Jaws', 'Spiderman']

and the movie String is 'Rocky' then don't modify the movies list

Print the message "Movie not found!"

If the movies list is null, or empty, print the message "Movie not found!"

Hint: check if the list is null or empty first.

\*/

// If the list is null or empty

if (movies == null || movies.isEmpty()) {

System.out.println("Movie not found!");

return;

}

// Remove the movie from the movies list

boolean removed = movies.removeIf(m -> m.equalsIgnoreCase(movie));

if (removed) {

System.out.println("Movie removed!");

} else {

System.out.println("Movie not found!");

}

}

public static String getRandomMovieFromWatchList(List<String> movies) {

/\*

Part 4.

TODO finish this method.

Return the name of a random movie from the movies list.

Don't modify the movies list.

If the movies list is null, or empty, return null.

Hint: check if the list is null or empty first.

\*/

"""

Returns a random movie from the provided watchlist.

Args:

watchlist (list): A list of movie titles.

Returns:

str: A randomly selected movie title from the watchlist.

"""

# Check if the watchlist is empty

if not watchlist:

return None

# Select a random index in the range of the watchlist

random\_index = random.randint(0, len(watchlist) - 1)

# Return the movie title at the random index

return watchlist[random\_index]

}

public static void printMoviesInNameOrder(List<String> movies) {

/\* Part 5.

TODO finish this method.

Print the movie names in alphabetical order, one movie per line.

\*\* Don't modify the original movies list! \*\*

If the movies list contains ['Up', 'Jaws', 'Spiderman'] you will

print

Jaws

Spiderman

Up

You should sort the movies using Java's default sort order for strings,

and print the exact text of the movie names from the list.

If the movies list contains ['Up', 'jaws', 'Spiderman']

Note 'jaws' has lowercase 'j' and lowercase letters are sorted after

uppercase letters.

you will print

Spiderman

Up

jaws

If the movies list is empty or null, print the message 'No movies'

This method will not return anything.

\*/

if (movies == null || movies.isEmpty()) {

System.out.println("No movies in list");

} else {

List<String> sortedMovies = new ArrayList<>(movies);

sortedMovies.sort(String.CASE\_INSENSITIVE\_ORDER);

for (String s : sortedMovies) {

System.out.println(s);

}

}

}

public static void printMoviesInWatchListOrder(List<String> movies) {

/\* Part 6.

TODO Print the movie names in watchlist order, one movie per line.

Include a number to indicate the movie's position in the watch list.

\*\* Don't modify the original movies list! \*\*

Don't print anything else - only the names of movies and the numbers.

If the movies list contains ['Up', 'Jaws', 'Spiderman'] you will print

1. Up

2. Jaws

3. Spiderman

If the movies list is empty or null, print the message 'No movies'

This method does not return anything.

\*/

if (movies == null || movies.isEmpty()) {

System.out.println("No movies in list");

} else {

for (String s : movies) {

System.out.println(s);

}

}

}

}