class FitnessCenterMembershipSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Using HashSet for unique member IDs

Set<String> hashSetMemberIds = new HashSet();

// Using TreeSet for unique member IDs

Set<String> treeSetMemberIds = new TreeSet();

while (true) {

System.out.println("\nFitness Center Membership System");

System.out.println("1. Add a new member (HashSet)");

System.out.println("2. Check member ID existence (HashSet)");

System.out.println("3. Display total number of members (HashSet)");

System.out.println("4. Add a new member (TreeSet)");

System.out.println("5. Check member ID existence (TreeSet)");

System.out.println("6. Display total number of members (TreeSet)");

System.out.println("7. Remove a member (HashSet)");

System.out.println("8. Remove a member (TreeSet)");

System.out.println("9. Display all member IDs (HashSet)");

System.out.println("10. Display all member IDs (TreeSet)");

System.out.println("11. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

// Add a new member using HashSet

System.out.print("Enter member ID to add (HashSet): ");

String hashSetMemberId = scanner.nextLine();

addMember(hashSetMemberId,);

break;

case 2:

// Check member ID existence using HashSet

System.out.print("Enter member ID to check (HashSet): ");

String hashSetCheckId = scanner.nextLine();

checkMemberExistence(hashSetCheckId, hashSetMemberIds);

break;

case 3:

// Display total number of members using HashSet

displayTotalMembers();

break;

case 4:

// Add a new member using TreeSet

System.out.print("Enter member ID to add (TreeSet): ");

String treeSetMemberId = scanner.nextLine();

addMember(treeSetMemberId, );

break;

case 5:

// Check member ID existence using TreeSet

System.out.print("Enter member ID to check (TreeSet): ");

String treeSetCheckId = scanner.nextLine();

checkMemberExistence(treeSetCheckId,);

break;

case 6:

// Display total number of members using TreeSet

displayTotalMembers(treeSetMemberIds);

break;

case 7:

// Remove a member using HashSet

System.out.print("Enter member ID to remove (HashSet): ");

String hashSetRemoveId = scanner.nextLine();

removeMember(hashSetRemoveId,);

break;

case 8:

// Remove a member using TreeSet

System.out.print("Enter member ID to remove (TreeSet): ");

String treeSetRemoveId = scanner.nextLine();

removeMember(treeSetRemoveId, treeSetMemberIds);

break;

case 9:

// Display all member IDs using HashSet

displayAllMembers(hashSetMemberIds);

break;

case 10:

// Display all member IDs using TreeSet

displayAllMembers(treeSetMemberIds);

break;

case 11:

// Exit the program

System.out.println("Exiting the program.");

scanner.close();

System.exit(0);

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

}

}

private static void addMember(String memberId, Set<String> memberIds) {

if (memberIds.add()) {

System.out.println("Member added successfully.");

} else {

System.out.println("Member ID already exists. Unable to add.");

}

}

private static void checkMemberExistence(String memberId, Set<String> memberIds) {

if (memberIds.contains(memberId)) {

System.out.println("Member ID ' + memberId + "' exists in the system.);

} else {

System.out.println("Member ID ' + memberId + "' does not exist in the system.);

}

}

private static void displayTotalMembers(Set<String> memberIds) {

System.out.println("Total number of members: " + memberIds.size());

}

private static void removeMember(String memberId, Set<String> memberIds) {

if (memberIds.remove()) {

System.out.println("Member removed successfully.");

} else {

System.out.println("Member ID + memberId + "' not found. Unable to remove.");

}

}

private static void displayAllMembers(Set<String> memberIds) {

System.out.println("List of all member IDs:");

for (String memberId : ) {

System.out.println(memberId);

}

}

}