Assignment-6

1.Create a program that uses an ArrayList to store a list of names. The program should allow the user to add and remove names from the list and should display the current list of names after each modification.

Code:

```java
package com.tecnotree.Assignment6;

import java.util.ArrayList;
import java.util.Scanner;

public class Question1 {
    public static void main(String[] args) {
        ArrayList<String> nameList = new ArrayList<String>();
        Scanner scanner = new Scanner(System.in);
        boolean exit = false;

        while (!exit) {
            System.out.println("Enter 1 to add a name, 2 to remove a name, or 3 to exit:");
            int choice = scanner.nextInt();
            scanner.nextLine(); // consume the newline character

            switch (choice) {
                case 1:
                    System.out.println("Enter a name to add:");
                    String name = scanner.nextLine();
                    nameList.add(name);
                    break;
                case 2:
                    System.out.println("Enter the index of the name to remove:");
                    int index = scanner.nextInt();
                    scanner.nextLine(); // consume the newline character
                    if (index >= 0 && index < nameList.size()) {
                        nameList.remove(index);
                    } else {
                        System.out.println("Invalid index!");
                    }
                    break;
                case 3:
                    exit = true;
                    break;
                default:
                    System.out.println("Invalid choice!");
            }

            System.out.println("Current list of names:");
            for (String n : nameList) {
                System.out.println(n);
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

https://codeshare.io/EBERWK

Output:

```
Enter 1 to add a name, 2 to remove a name, or 3 to quit:
1
Enter a name to add:
nishanth
Current list of names: [nishanth]
Enter 1 to add a name, 2 to remove a name, or 3 to quit:
2
Enter the name to remove:
nishanth
Current list of names: []
Enter 1 to add a name, 2 to remove a name, or 3 to quit:
```

2.Create a program that uses a HashMap to store a dictionary of words and their meanings. The program should allow the user to add new words and meanings, and should display the meaning of a word when the user enters the word.

```java
package com.tecnotree.Assignment6;

import java.util.HashMap;
import java.util.Scanner;

public class Question2 {
    public static void main(String[] args) {
        HashMap<String, String> dictionary = new HashMap<String, String>();
        Scanner scanner = new Scanner(System.in);
        boolean exit = false;

        while (!exit) {
            System.out.println("Enter 1 to add a word, 2 to display the meaning of a word, or 3 to exit:");
            int choice = scanner.nextInt();
            scanner.nextLine(); // consume the newline character

            switch (choice) {
                case 1:
                    System.out.println("Enter a word to add:");
                    String word = scanner.nextLine();
                    System.out.println("Enter the meaning of the word:");
                    String meaning = scanner.nextLine();
                    dictionary.put(word, meaning);
                    break;
                case 2:
                    System.out.println("Enter a word to display its meaning:");
                    word = scanner.nextLine();
                    meaning = dictionary.get(word);
                    if (meaning != null) {
                        System.out.println("Meaning: " + meaning);
                    } else {
                        System.out.println("Word not found in dictionary!");
                    }
                    break;
                case 3:
                    exit = true;
                    break;
                default:
                    System.out.println("Invalid choice!");
            }

            System.out.println();
        }

        scanner.close();
    }
}
```

https://codeshare.io/RbvpAB

Output:

```
Enter 1 to add a word, 2 to display the meaning of a word, or 3 to exit:
1
Enter a word to add:
Happy
Enter the meaning of the word:
Enjoying or characterized by well-being and contentment

Enter 1 to add a word, 2 to display the meaning of a word, or 3 to exit:
3
```

3.Create a program that uses a TreeSet to store a list of integers. The program should allow the user to add and remove integers from the set, and should display the current set of integers after each modification.

```java
package com.tecnotree.Assignment6;

import java.util.Scanner;
import java.util.TreeSet;

public class Question3 {
    public static void main(String[] args) {
        TreeSet<Integer> integerSet = new TreeSet<Integer>();
        Scanner scanner = new Scanner(System.in);
        boolean exit = false;

        while (!exit) {
            System.out.println("Enter 1 to add an integer, 2 to remove an integer, or 3 to exit:");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Enter an integer to add:");
                    int number = scanner.nextInt();
                    integerSet.add(number);
                    break;
                case 2:
                    System.out.println("Enter an integer to remove:");
                    number = scanner.nextInt();
                    if (integerSet.contains(number)) {
                        integerSet.remove(number);
                    } else {
                        System.out.println("Integer not found in set!");
                    }
                    break;
                case 3:
                    exit = true;
                    break;
                default:
                    System.out.println("Invalid choice!");
            }

            System.out.println("Current set of integers:");
            for (int n : integerSet) {
                System.out.print(n + " ");
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

https://codeshare.io/JbMRA6

Output:

```
Enter 1 to add an integer, 2 to remove an integer, or 3 to exit:
1
Enter an integer to add:
7
Current set of integers:
7
Enter 1 to add an integer, 2 to remove an integer, or 3 to exit:
1
Enter an integer to add:
151
Current set of integers:
7 151
Enter 1 to add an integer, 2 to remove an integer, or 3 to exit:
2
Enter an integer to remove:
151
Current set of integers:
7
Enter 1 to add an integer, 2 to remove an integer, or 3 to exit:
3
Current set of integers:
7
```

4.Create a program that uses a LinkedList to implement a queue. The program should allow the user to add and remove items from the queue, and should display the current contents of the queue after each modification.

```java
package com.tecnotree.Assignment6;

import java.util.LinkedList;
import java.util.Scanner;

public class Question4 {
    public static void main(String[] args) {
        LinkedList<String> queue = new LinkedList<String>();
        Scanner scanner = new Scanner(System.in);
        boolean exit = false;

        while (!exit) {
            System.out.println("Enter 1 to add an item to the queue, 2 to remove an item, or 3 to exit:");
            int choice = scanner.nextInt();
            scanner.nextLine(); // consume the newline character

            switch (choice) {
                case 1:
                    System.out.println("Enter an item to add to the queue:");
                    String item = scanner.nextLine();
                    queue.add(item);
                    break;
                case 2:
                    if (queue.isEmpty()) {
                        System.out.println("Queue is empty!");
                    } else {
                        System.out.println("Removed item: " + queue.remove());
                    }
                    break;
                case 3:
                    exit = true;
                    break;
                default:
                    System.out.println("Invalid choice!");
            }

            System.out.println("Current contents of the queue:");
            for (String s : queue) {
                System.out.print(s + " ");
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

https://codeshare.io/YLEXBb

Output:

```
Enter 1 to add an item to the queue, 2 to remove an item, or 3 to exit:
1
Enter an item to add to the queue:
red
Current contents of the queue:
red
Enter 1 to add an item to the queue, 2 to remove an item, or 3 to exit:
1
Enter an item to add to the queue:
blue
Current contents of the queue:
red blue
Enter 1 to add an item to the queue, 2 to remove an item, or 3 to exit:
2
Removed item: red
Current contents of the queue:
blue
Enter 1 to add an item to the queue, 2 to remove an item, or 3 to exit:
3
Current contents of the queue:
blue
```

5.Create a program that uses a HashSet to store a set of strings. The program should read in a text file, and should add each word in the file to the set of strings. After all words have been added, the program should display the number of unique words in the file.

```java
package com.tecnotree.Assignment6;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.HashSet;
import java.util.Scanner;

public class Question5 {
    public static void main(String[] args) {
        HashSet<String> words = new HashSet<String>();

        try {
            File file = new File("C:\\Users\\fawazmo\\eclipse-workspace\\Assignment6\\src\\com\\tecnotree\\Assigment6\\input.txt");
            Scanner scanner = new Scanner(file);

            while (scanner.hasNext()) {
                String word = scanner.next();
                words.add(word);
            }

            scanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found!");
            System.exit(1);
        }

        System.out.println("Number of unique words in the file: " + words.size());
    }
}
```

https://codeshare.io/BA7RbO

Output:

```
Number of unique words in the file: 15
```