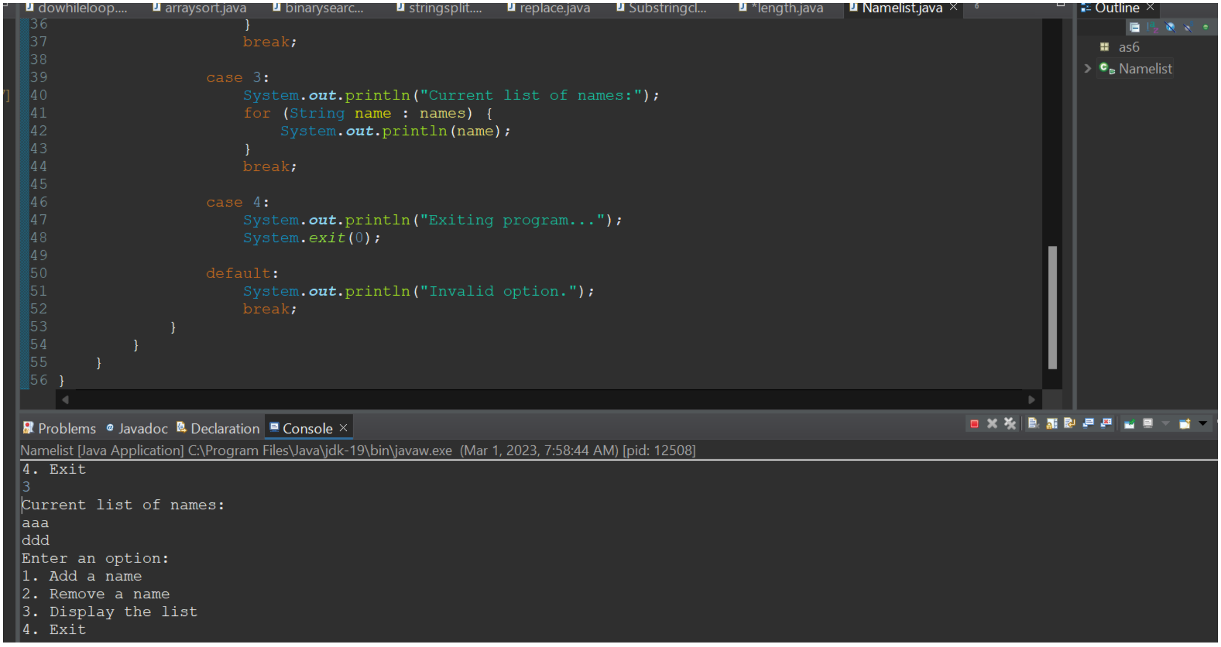
Assignment-06

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: https://codeshare.io/bvOynb



```
                    }
                    break;

                case 3:
                    System.out.println("Current list of names:");
                    for (String name : names) {
                        System.out.println(name);
                    }
                    break;

                case 4:
                    System.out.println("Exiting program...");
                    System.exit(0);

                default:
                    System.out.println("Invalid option.");
                    break;
            }
        }
    }
}
```

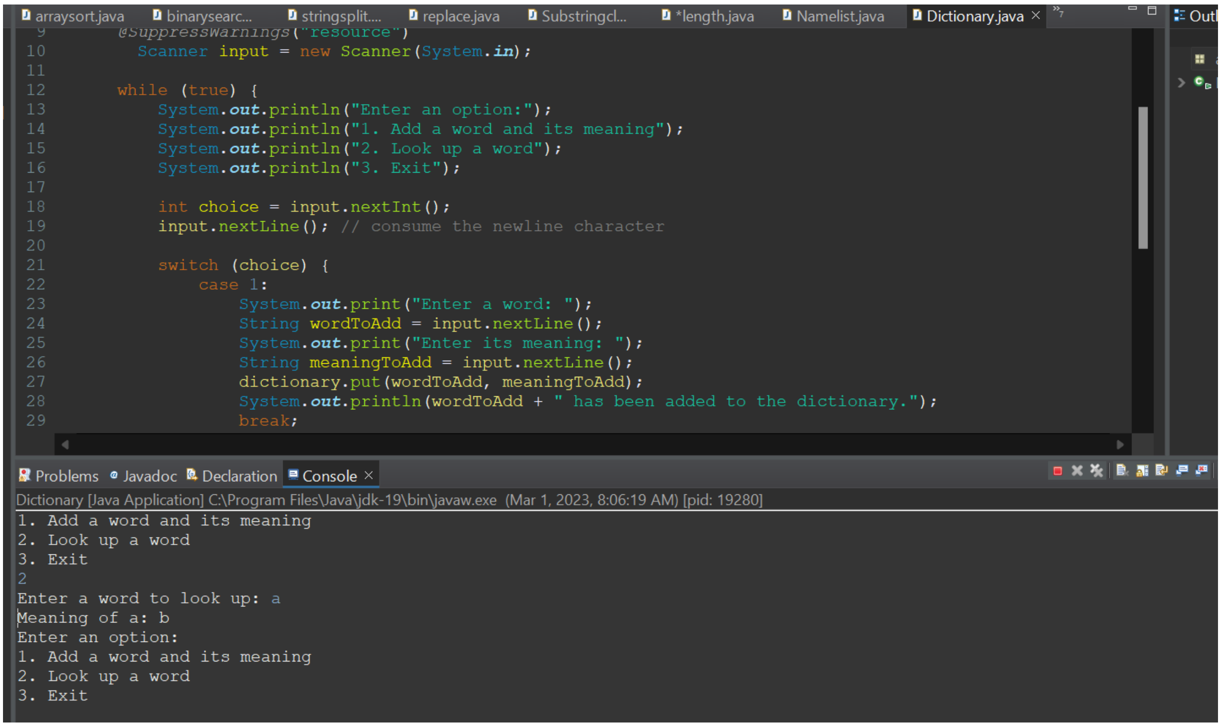
Namelist [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Mar 1, 2023, 7:58:44 AM) [pid: 12508]

```
4. Exit
3
Current list of names:
aaa
ddd
Enter an option:
1. Add a name
2. Remove a name
3. Display the list
4. Exit
```

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.

Code: https://codeshare.io/PdE0dP

V



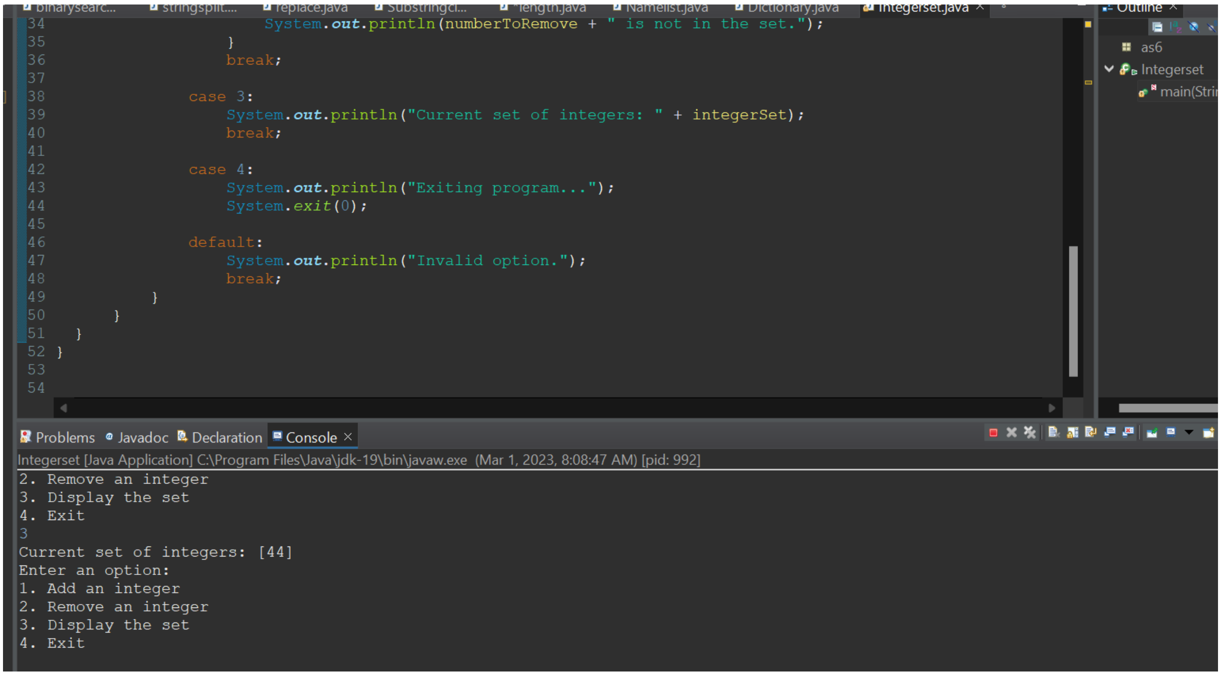
```java
        @SuppressWarnings("resource")
          Scanner input = new Scanner(System.in);

        while (true) {
            System.out.println("Enter an option:");
            System.out.println("1. Add a word and its meaning");
            System.out.println("2. Look up a word");
            System.out.println("3. Exit");

            int choice = input.nextInt();
            input.nextLine(); // consume the newline character

            switch (choice) {
                case 1:
                    System.out.print("Enter a word: ");
                    String wordToAdd = input.nextLine();
                    System.out.print("Enter its meaning: ");
                    String meaningToAdd = input.nextLine();
                    dictionary.put(wordToAdd, meaningToAdd);
                    System.out.println(wordToAdd + " has been added to the dictionary.");
                    break;
```

Dictionary [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Mar 1, 2023, 8:06:19 AM) [pid: 19280]

```
1. Add a word and its meaning
2. Look up a word
3. Exit
2
Enter a word to look up: a
Meaning of a: b
Enter an option:
1. Add a word and its meaning
2. Look up a word
3. Exit
```

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.

Code: https://codeshare.io/pqkjqx



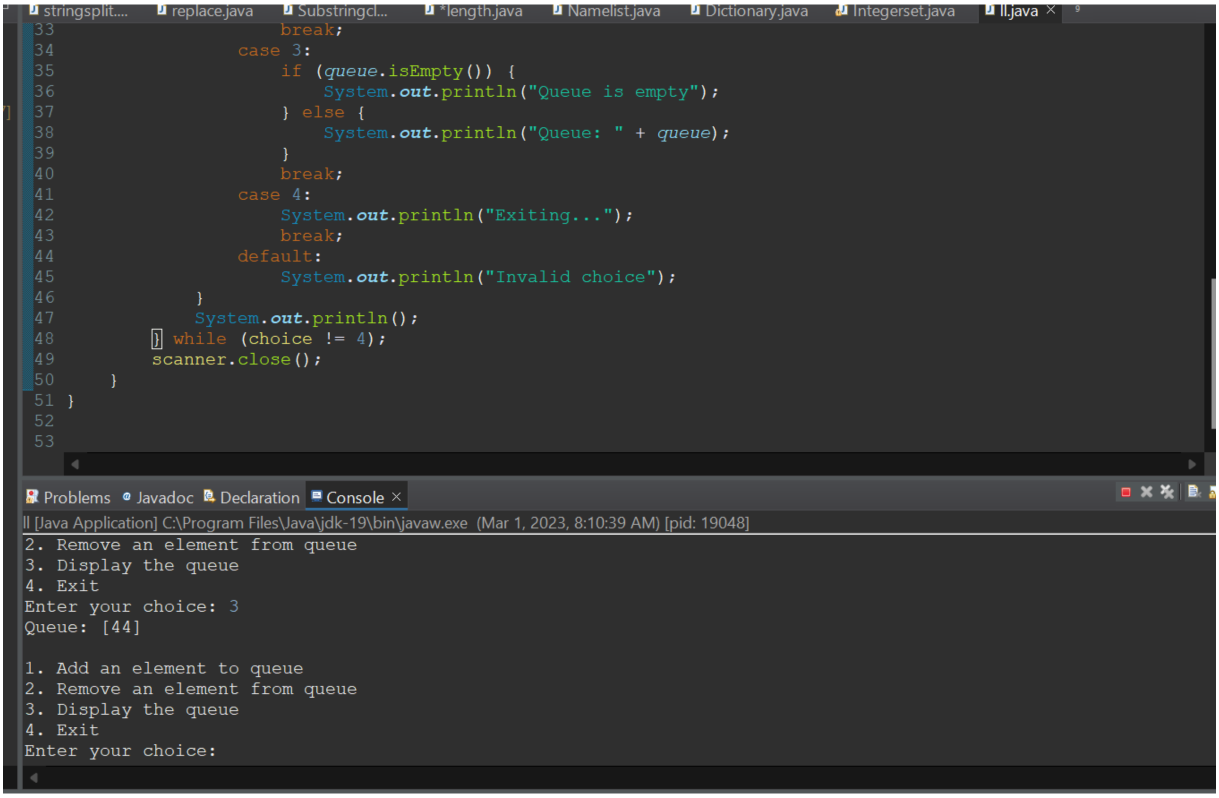
```
                      System.out.println(numberToRemove + " is not in the set.");
                  }
                  break;

              case 3:
                  System.out.println("Current set of integers: " + integerSet);
                  break;

              case 4:
                  System.out.println("Exiting program...");
                  System.exit(0);

              default:
                  System.out.println("Invalid option.");
                  break;
          }
      }
  }
}
```

Integerset [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Mar 1, 2023, 8:08:47 AM) [pid: 992]

```
2. Remove an integer
3. Display the set
4. Exit
3
Current set of integers: [44]
Enter an option:
1. Add an integer
2. Remove an integer
3. Display the set
4. Exit
```

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.

Code: https://codeshare.io/dwQywZ



```
                    break;
                case 3:
                    if (queue.isEmpty()) {
                        System.out.println("Queue is empty");
                    } else {
                        System.out.println("Queue: " + queue);
                    }
                    break;
                case 4:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice");
            }
            System.out.println();
        } while (choice != 4);
        scanner.close();
    }
}
```

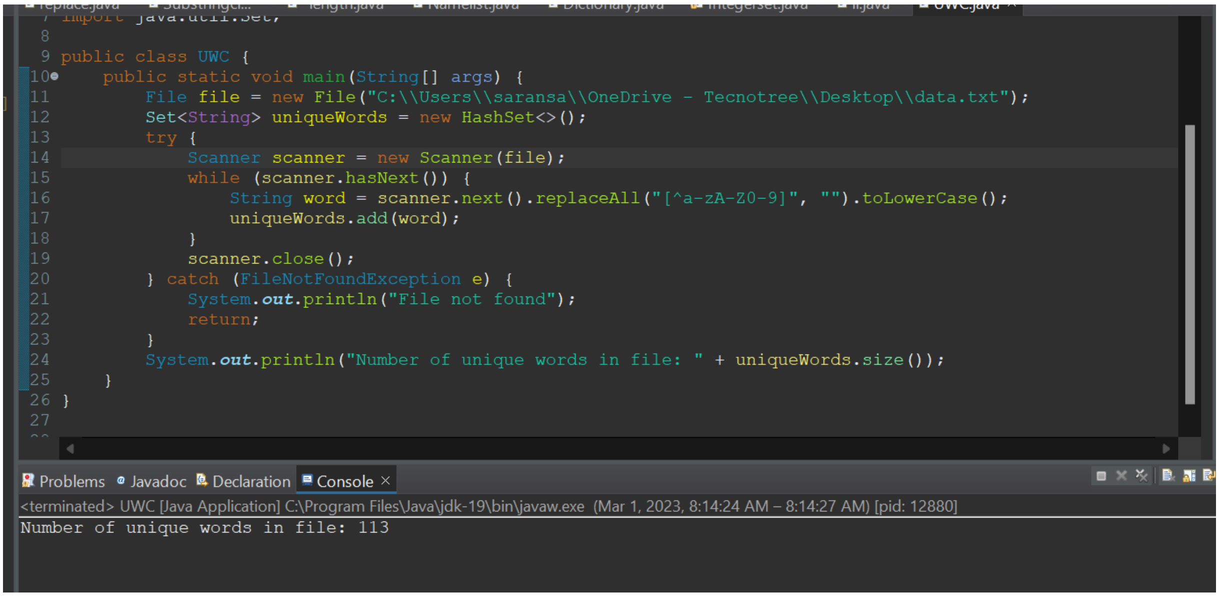
ll [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Mar 1, 2023, 8:10:39 AM) [pid: 19048]

```
2. Remove an element from queue
3. Display the queue
4. Exit
Enter your choice: 3
Queue: [44]

1. Add an element to queue
2. Remove an element from queue
3. Display the queue
4. Exit
Enter your choice:
```

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.

Code: https://codeshare.io/bvOyvbs



```
import java.util.Set;

public class UWC {
    public static void main(String[] args) {
        File file = new File("C:\\Users\\saransa\\OneDrive - Tecnotree\\Desktop\\data.txt");
        Set<String> uniqueWords = new HashSet<>();
        try {
            Scanner scanner = new Scanner(file);
            while (scanner.hasNext()) {
                String word = scanner.next().replaceAll("[^a-zA-Z0-9]", "").toLowerCase();
                uniqueWords.add(word);
            }
            scanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found");
            return;
        }
        System.out.println("Number of unique words in file: " + uniqueWords.size());
    }
}
```

```
<terminated> UWC [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (Mar 1, 2023, 8:14:24 AM – 8:14:27 AM) [pid: 12880]
Number of unique words in file: 113
```