HASH TABLE – LAB 05

EC 4070

DATA STRUCTURES AND ALGORITHMS

NAME : **WIJAYAWARDHANA W.A.H.A.**

REGISTRATION NO. : **2019/E/166**

SEMESTER : **SEMESTER 04**

DATE ASSIGNED : **23 MARCH 2022**

01.

Code:

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

/**
 *  MaximumOccurrence method use for store the char values in array list.
 */
public class MaximumOccurrence {
   ArrayList<Character> HashTableElement = new ArrayList<Character>();
   int countingArray[][] = new int[128][2];
   Scanner scanner = new Scanner(System.in);
   String inputString;
   // Default constructor.
   public void setElements()
   {
      System.out.println("Enter the word : ");
      inputString = scanner.nextLine();
      for(int k =0; k<128;k++)
      {
         countingArray[k][0] = k;
      }

   }

   /**
    * setHashTableElement method use for set elements in hash table.
    */
   // setHashTableElement method for setting HashTableElement array list.
   public void setHashTableElement()
   {
      for(int i =0; i<inputString.length(); i++)  // Adding element by element into array list.
      {
         HashTableElement.add(i,inputString.charAt(i));
         inputString.indexOf(i);
      }
   }

   /**
    * customHashTable method use to add the char values to hash table according to the ascii
value.
    */
   public void customHashTable()
   {
      for(int i =0; i<HashTableElement.size(); i++)
```

```
        {
            int tempValue = HashTableElement.get(i)%127;
            countingArray[tempValue][1] = countingArray[tempValue][1]+1;
        }
    }

    /**
     * findMaximum method use to find the maximum number of element
     */
    public void findMaximum()
    {
        int maximum = countingArray[0][1];
        int maximumIndex = countingArray[0][0];
        for(int i =0; i<128;i++)
        {
            if(maximum < countingArray[i][1])
            {
                maximumIndex = countingArray[i][0];
                maximum = countingArray[i][1];
            }
        }
        char tempChar = (char) countingArray[maximumIndex][0];
        int temp = countingArray[maximum][0];
        System.out.println("Maximum occurrence : ");
        System.out.print(tempChar + "  "+temp);
    }

    /**
     * main method use for create objects and calling methods.
     * @param args
     */
    public static void main(String[] args) {
        MaximumOccurrence newObject = new MaximumOccurrence();
        newObject.setElements();
        newObject.setHashTableElement();
        newObject.customHashTable();
        newObject.findMaximum();
    }
}
```

Output:

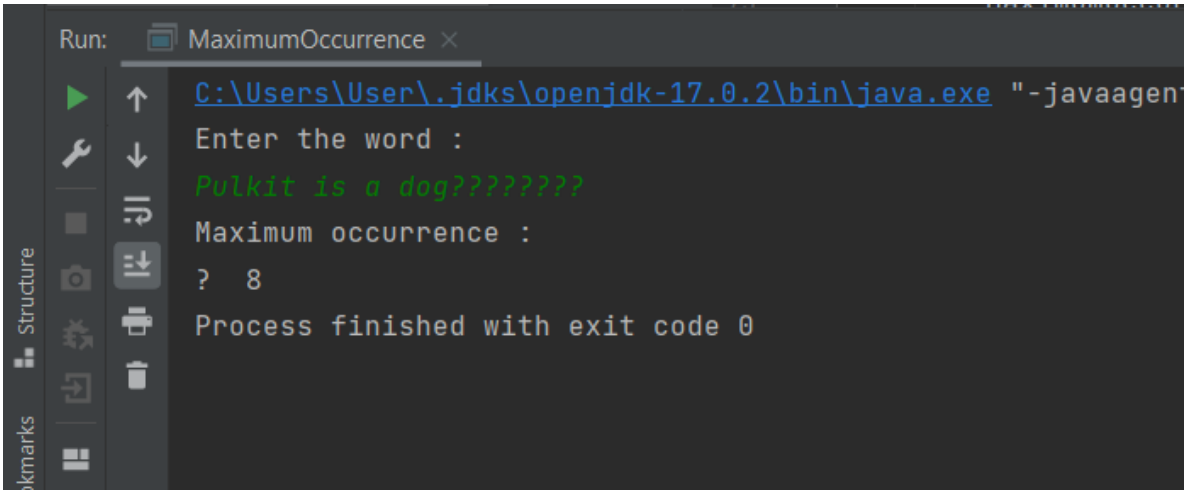

FIGURE 01 – OUTPUT

02.

Code:

```
package favoritegame;
import java.util.*;

/**
 *
 * @author 2019e166
 */
public class FavoriteGame {
   int numberOfEntries;
   HashMap<String,String> hashtable = new HashMap<String,String>(numberOfEntries);
   HashMap<String,Integer> hashtableCount = new HashMap<String,Integer>(numberOfEntries);
   Scanner scanner = new Scanner(System.in);

   /**
    * @setNumberOfEntries is for setting the number of entries.
    */
   public void setNumberOfEntries()
   {
      System.out.print("Enter number of entries : ");
      numberOfEntries = scanner.nextInt();
   }

   /**
    * @setHashtable use for setting elements to hash map.
    */
   public void setHashmap()
   {
      for (int i =0; i<numberOfEntries; i++) {
```

```
            String nameString = scanner.next();
            String gameString = scanner.next();
            hashtable.put(nameString,gameString);
        }
    }

    /**
     * Count the elements and add them to hash map.
     */

    public void countElement()
    {
        for(String elements : hashtable.values())
        {
            if(hashtableCount.containsKey(elements))
            {
                int count = hashtableCount.get(elements);
                count= count+1;
                hashtableCount.put(elements,count);
            }
            else
            {
                hashtableCount.put(elements,1);
            }
        }
    }

    /**
     * Find the maximum count of the favorite game and football count.
     */

    public void findMax()
    {
        int maxCount = 0;
        String maxLikeGameName = " ";
        for(String element : hashtableCount.keySet())
        {
            int gameCount = hashtableCount.get(element);
            if(maxCount <= gameCount)
            {
                maxCount = gameCount;
                maxLikeGameName = element;
            }
        }
        System.out.println(maxLikeGameName);
        System.out.println("Football : " + hashtableCount.get("football"));

    }
```

```
    /**
     * @printFootballCount use to get number of people who likes football.
     */


    /**
     * @param args the command line arguments
     * @main for creating an object and calling methods.
     */
    public static void main(String[] args) {
        FavoriteGame favoriteGame = new FavoriteGame();
        favoriteGame.setNumberOfEntries();
        favoriteGame.setHashmap();
        favoriteGame.countElement();
        favoriteGame.findMax();
    }

}
```

Output:

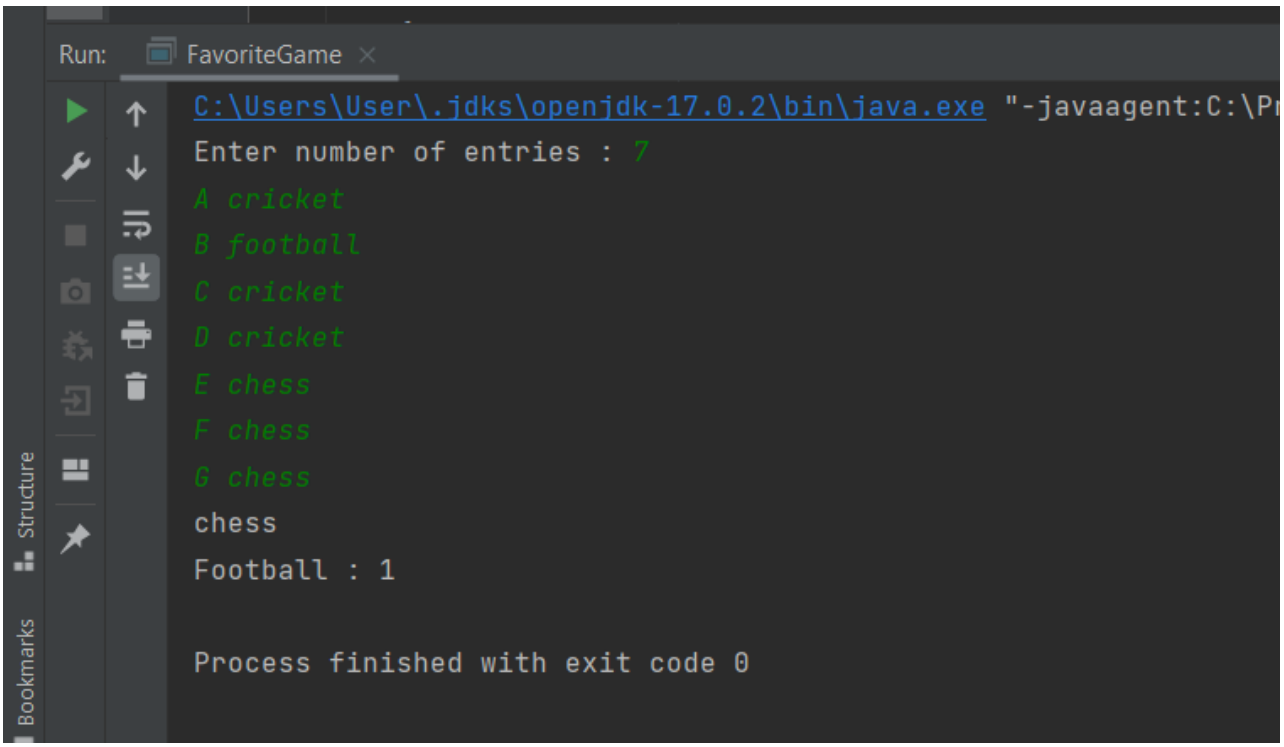

FIGURE 02 - OUTPUT