

# Programming Assignment II

Hashing

1) E X A M Q U ~~ESTION~~,  $m=5$ ,  $||k\%m$

Input  
Hash  
Table

1. E (5th)  
 $|| (5) \% 5 = 0$

0	1	2	3	4
E				

2. X (24th)  
 $|| (24) \% 5 = 4$

0	1	2	3	4
E				X

3. A (1st)  
 $|| (1) \% 5 = 1$

0	1	2	3	4
E	A			X

4. M (13th)  
 $|| (13) \% 5 = 3$

0	1	2	3	4
E	A		M	X

5. Q (17th)  
 $|| (17) \% 5 = 2$

0	1	2	3	4
E	A	Q	M	X

6. U (21st)  
 $|| (21) \% 5 = 1$

0	1	2	3	4
E	A	Q	M	X
	U			

7. E (5th)  
repeat letter

8. S (19th)  
 $|| (19) \% 5 = 4$

0	1	2	3	4
E	A	Q	M	X
	U		S	

9. T (20th)  
 $|| (20) \% 5 = 0$

0	1	2	3	4
E	A	Q	M	X
T	U		S	

10. I (9th)  
 $|| (9) \% 5 = 4$

0	1	2	3	4
E	A	Q	M	X
	T		S	
			I	

11. O (15th)  
 $|| (15) \% 5 = 0$

0	1	2	3	4
E	A	Q	M	X
T	U		S	
O			I	

12. N (14th)  
 $|| (14) \% 5 = 4$

0	1	2	3	4
E	A	Q	M	X
T	U		S	
O			I	
			N	

## 2) EXAM QUESTION, $m=16$ , $11k \% M$

Input hash table	E (5th) $11(5) \% 16 = 7$ <table border="1"> <tr><td>7</td></tr> <tr><td>E</td></tr> </table>	7	E	X (24th) $11(24) \% 16 = 8$ <table border="1"> <tr><td>7 8</td></tr> <tr><td>EX</td></tr> </table>	7 8	EX	A (1st) $11(1) \% 16 = 11$ <table border="1"> <tr><td>7 8 ... 11</td></tr> <tr><td>EX ... A</td></tr> </table>	7 8 ... 11	EX ... A	M (13th) $11(13) \% 16 = 15$ <table border="1"> <tr><td>7 8 ... 11 ... 15</td></tr> <tr><td>EX ... A ... M</td></tr> </table>	7 8 ... 11 ... 15	EX ... A ... M
	7											
	E											
7 8												
EX												
7 8 ... 11												
EX ... A												
7 8 ... 11 ... 15												
EX ... A ... M												
Q (17th) $11(17) \% 16 = 11$ <table border="1"> <tr><td>7 8 - 11 12 - 15</td></tr> <tr><td>EX - A Q - M</td></tr> </table>	7 8 - 11 12 - 15	EX - A Q - M	U (21st) $11(21) \% 16 = 7$ <table border="1"> <tr><td>7 8 9 11 12 15</td></tr> <tr><td>EXU A Q M</td></tr> </table>	7 8 9 11 12 15	EXU A Q M	S (19th) $11(19) \% 16 = 1$ <table border="1"> <tr><td>1 7 8 9 11 12 15</td></tr> <tr><td>S EXU A Q M</td></tr> </table>	1 7 8 9 11 12 15	S EXU A Q M	T (20th) $11(20) \% 16 = 12$ <table border="1"> <tr><td>1 7 8 9 11 12 13 15</td></tr> <tr><td>S EXU A Q T M</td></tr> </table>	1 7 8 9 11 12 13 15	S EXU A Q T M	
7 8 - 11 12 - 15												
EX - A Q - M												
7 8 9 11 12 15												
EXU A Q M												
1 7 8 9 11 12 15												
S EXU A Q M												
1 7 8 9 11 12 13 15												
S EXU A Q T M												
I (9th) $11(9) \% 16 = 3$ <table border="1"> <tr><td>1 3 7 8 9 11 12 13 15</td></tr> <tr><td>S I EXU A Q T M</td></tr> </table>	1 3 7 8 9 11 12 13 15	S I EXU A Q T M	O (15th) $11(15) \% 16 = 5$ <table border="1"> <tr><td>1 3 5 7 8 9 11 12 13 15</td></tr> <tr><td>S I O EXU A Q T M</td></tr> </table>	1 3 5 7 8 9 11 12 13 15	S I O EXU A Q T M	N (14th) $11(14) \% 16 = 10$ <table border="1"> <tr><td>1 3 5 7 8 9 10 11 12 13 15</td></tr> <tr><td>S I O EXU N A Q T M</td></tr> </table>	1 3 5 7 8 9 10 11 12 13 15	S I O EXU N A Q T M				
1 3 7 8 9 11 12 13 15												
S I EXU A Q T M												
1 3 5 7 8 9 11 12 13 15												
S I O EXU A Q T M												
1 3 5 7 8 9 10 11 12 13 15												
S I O EXU N A Q T M												

Code output (INCLUDED IN SPELLCHECKERAPP.JAVA):

```

    try
    {
        // Create a new instance of SpellChecker, loading the dictionary from "words.txt"
        SpellChecker spellChecker = new SpellChecker(wordlistPath:"words.txt");
        // Run the spell checker
        runSpellChecker(spellChecker);
    }
    catch (IOException e)
    {
        // Print an error message if the dictionary file failed to load
        System.err.println("Failed to load dictionary file: " + e.getMessage());
    }
}

```

```

// Method to run the spell checker
private static void runSpellChecker(SpellChecker spellChecker)
{
    // Create a scanner object to read user input
    Scanner inputScanner = new Scanner(System.in);
    String inputWord;

    // Repeat until the user types 'exit'
    do
    {
        // Prompt the user to enter a word to check
        System.out.print("\nEnter a word to check, or type 'exit' to quit: ");
        // Read the input word and remove any leading or trailing whitespace
        inputWord = inputScanner.nextLine().trim();

        // If the input word is not 'exit', process it
        if (!inputWord.equalsIgnoreCase("exit"))
        {
            processInput(spellChecker, inputWord);
        }
    } while (!inputWord.equalsIgnoreCase("exit"));

    // Close the scanner
    inputScanner.close();
}

```

```

// Method to process user input and check the spelling
private static void processInput(SpellChecker spellChecker, String word)
{
    // If the input word is in the dictionary, it is spelled correctly
    if (spellChecker.isValidWord(word))
    {
        System.out.println("\nNo mistakes found.");
    }
    else
    {
        // If the input word is not in the dictionary, it is misspelled
        System.out.println("\nThe word is misspelled.");
        // Get suggestions for corrections
        Set<String> suggestions = spellChecker.suggestCorrections(word);
    }
}

```

```

// SpellChecker.java
import java.io.*;
import java.util.*;

public class SpellChecker {
    private List<String> dictionary;

    public SpellChecker(String wordlistPath) {
        try {
            dictionary = new ArrayList<String>();
            BufferedReader reader = new BufferedReader(new FileReader(wordlistPath));
            String word;
            while ((word = reader.readLine()) != null) {
                dictionary.add(word.toLowerCase());
            }
            reader.close();
        } catch (IOException e) {
            System.err.println("Failed to load dictionary file: " + e.getMessage());
        }
    }

    public boolean isValidWord(String word) {
        return dictionary.contains(word.toLowerCase());
    }

    public Set<String> suggestCorrections(String word) {
        Set<String> suggestions = new HashSet<String>();
        // Implement Levenshtein distance algorithm here
        // For simplicity, we'll use a placeholder set of suggestions
        suggestions.add("algorithm");
        suggestions.add("algorithms");
        suggestions.add("aalgorithms");
        suggestions.add("algorithmm");
        suggestions.add("algorithn");
        suggestions.add("algoriithm");
        suggestions.add("algoriithms");
        suggestions.add("saadadadasdasdasdaw");
        return suggestions;
    }
}

```

```

Enter a word to check, or type 'exit' to quit: apple
No mistakes found.
Enter a word to check, or type 'exit' to quit: watermelo
The word is misspelled.
Suggested words:
    watermelon
Enter a word to check, or type 'exit' to quit: lgorithm
The word is misspelled.
Suggested words:
    algorithm
Enter a word to check, or type 'exit' to quit: lgorithms
The word is misspelled.
Suggested words:
    algorithms
Enter a word to check, or type 'exit' to quit: aalgorithm
The word is misspelled.
Suggested words:
    algorithm
Enter a word to check, or type 'exit' to quit: algorithmm
The word is misspelled.
Suggested words:
    algorithm
Enter a word to check, or type 'exit' to quit: algoriithm
The word is misspelled.
Suggested words:
    algorithm
Enter a word to check, or type 'exit' to quit: algoriithms
The word is misspelled.
Suggested words:
    algorithms
Enter a word to check, or type 'exit' to quit: saadadadasdasdasdaw
The word is misspelled.
No suggestions available.
Enter a word to check, or type 'exit' to quit: 

```