CPSC 1150 – Program Design

Assignment #4

Total Marks: 54 marks (4 marks are bonus marks)

Goals

This assignment gives students more experience on:

- Problem Solving (especially with methods, Arrays, command line arguments)
- Writing Algorithms, using flowchart or pseudocode
- Testing Algorithms
- Writing Java programs

Problem Description

Exercise 1: [32 marks] Write a java program named **ArraysExercise.java** including the following methods:

1. **[2 marks]** Write a method named **getArray** that asks user to input the array size n, then initializes an array of n integers from user input and returns the array.
2. **[2 marks]** Write a method named **printArray** that given an array of integers, prints the array in a tabular format with 5 number in each line.
3. **[3 marks]** Write a method named **findMax** that given an array of integers finds and returns the maximum value in the array.
4. **[3 marks]** Write a method named **findMin** that given an array of integers finds and returns the minimum value in the array.
5. **[3 marks]** Write a method named **findMaxIndex** that given an array of integers finds and returns the first index of maximum value in the array.
6. **[3 marks]** Write a method named **findMinIndex** that given an array of integers finds and returns the last index of minimum value in the array.
7. **[3 marks]** Write a method named **isSortedAscend** that given an array of integers, it checks whether array is sorted in ascending order, and returns true or false correspondingly.
8. **[3 marks]** Write a method named **isSortedDescend** that given an array of integers, it checks whether array is sorted in descending order, and returns true or false correspondingly.
9. **[5 marks]** Write a method named **swapNeighbor** that given an array, it compares every two neighbor numbers in the array (eg. A[i] and A[i+1]) and swaps them if A[i] is more than A[i+1]. It finally passes the new changes to the caller method. This method should be void method.
   For example, for the given array {12, 9, 15, 7, 3}, as the result of calling swapNeighbor, the array must be changed to {9, 12, 7, 3, 15}.
10. **[5 marks]** Write a method named **merge** that given two arrays sorted in ascending order, it merges them into one sorted array in ascending order, and returns the merged array.
    For example, if the given arrays are {2, 6, 9} and {-1, 5, 11, 12}, then the merged array must be {-1, 2, 5, 6, 9, 11, 12}. DO NOT use any sort algorithm.

    To test your program, use the following idea for your main method. Test each function alone. This way you are sure that function is working well and is ready to be used along with other functions in your program.

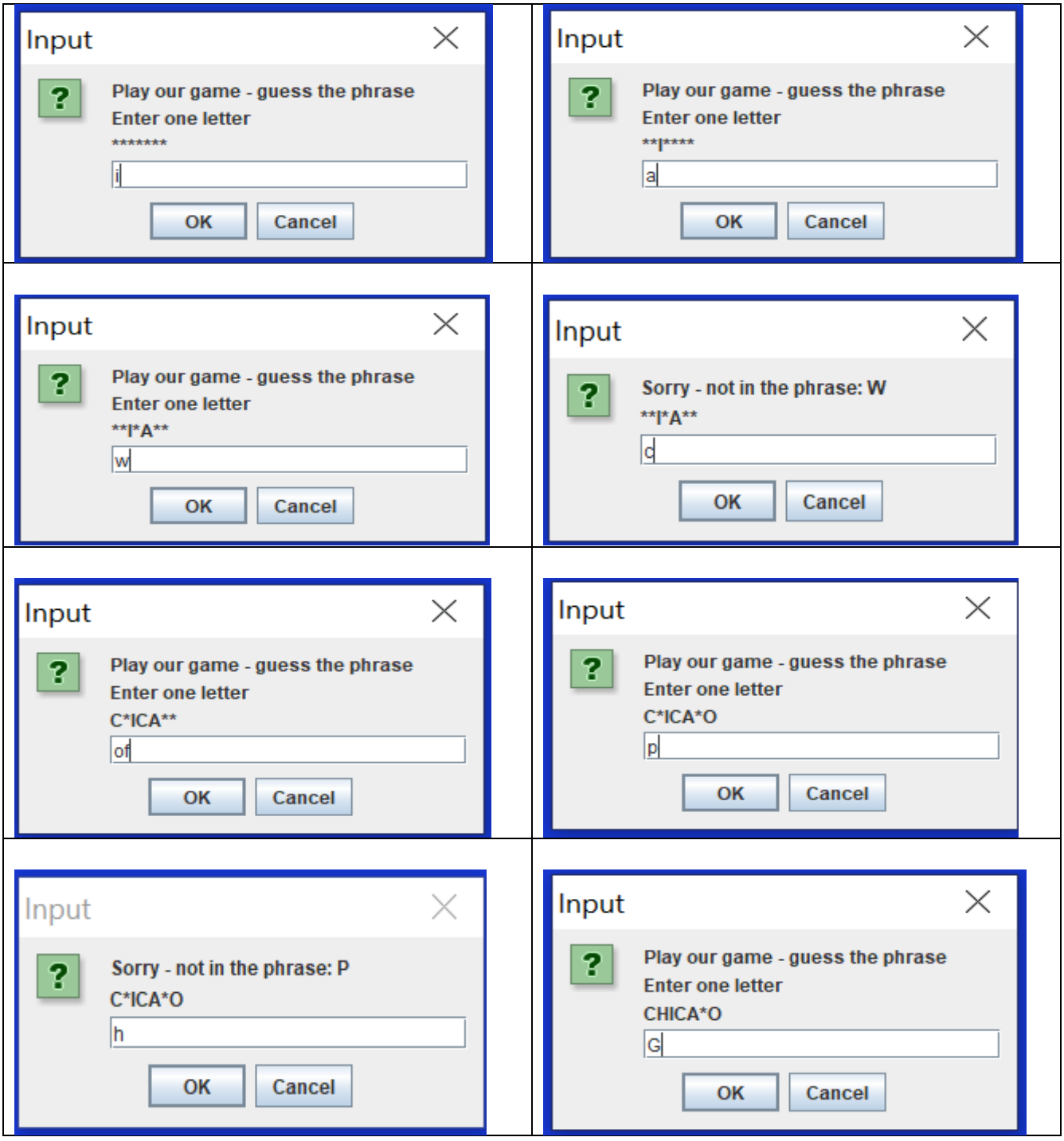
```
public class ArraysExercise{
    public static void main(String [] args){
        // call getArray and create an array named arr1
        // call printArray given arr1
        // call findMax given arr1
        // print the return value of findMax
        // call findMin given arr1
        // print the return value of findMin
        // call findMaxIndex given arr1
        // print the return value of findMaxIndex
        // call findMinIndex given arr1
        // print the return value of findMinIndex
        // call isSortedAscend given {2, 5, 7, 9} and print the result
        // call isSortedAscend given {2, 15, 7, 29} and print the result
        // call isSortedDescend given {2, 5, 7, 9} and print the result
        // call isSortedDescend given {25, 15, 7, 2} and print the result
        // call swapNeighbor given arr1
        // print the result of calling swapNeighbor given arr1
        // call merge given {2, 6, 9} and {-1, 5, 11, 12} and print the result
        // call getArray and create an array named arr2
        // call merge given arr1 and arr2
    }
}
```

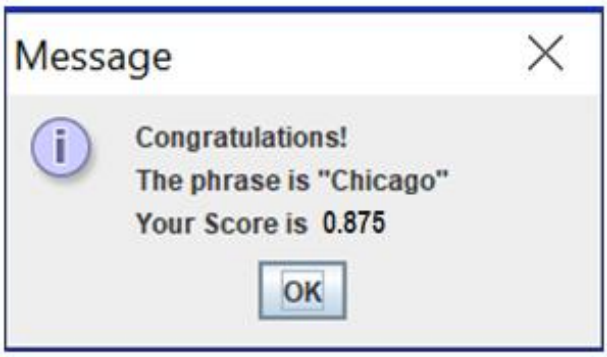
Exercise 2 – Secret Phrase Game

[22 marks]: Create a guessing game like Hangman, in which the user guesses letters, and then attempts to guess a partially hidden phrase. Display a phrase in which some of the letters are replaced by asterisk: for example, "G* T***" for "`Go Team`". Each time a user guesses a letter, according to the guess, either place the letter in the correct spot(s) in the phrase and display it to user or tell the user the guessed letter is not in the phrase. Display a congratulatory message when the entire correct phrase has been deduced. Save the game as **SecretPhrase.java**.

- The phrase to be guessed is selected randomly from a list of at least 10 phrases.
- The clue is presented to the user with asterisks replacing letters to be guessed but with space in the appropriate locations. For example, if the phrase to be guessed is *`No man is an island`*, then the user sees the following as first clue: `** *** ** ** ******`
  The spaces provide valuable information as where individual words start and end.
- Make sure that when a user makes a correct guess, all the matching letters are filled in, regardless of case.
- Use `javax.swing.JOptionPane` class to have a graphical user interface (GUI) (Optional – 4 bonus marks).
- The guessing should be case-insensitive.
- If user enters more than 1 letters, only the first letter would be considered as the guess.
- At the end of round, it shows the score of users which is calculated by the length of phrase divided by the number of user's guesses. For example, if the phrase to be guessed is "The `Wizard of Oz`" and user have guessed it after 13 guesses, the score is 1.23 (i.e. 16/13). Obviously, user get higher score if (s)he can guess the phrase with less guesses.

The following show a sample run of program (images are from left to right):


Input
Play our game - guess the phrase
Enter one letter
****
i
OK
Cancel
Input
Play our game - guess the phrase
Enter one letter
I**
a
OK
Cancel
Input
Play our game - guess the phrase
Enter one letter
I*A
w
OK
Cancel
Input
Sorry - not in the phrase: W
I*A
c
OK
Cancel
Input
Play our game - guess the phrase
Enter one letter
C*ICA**
of
OK
Cancel
Input
Play our game - guess the phrase
Enter one letter
C*ICA*O
p
OK
Cancel
Input
Sorry - not in the phrase: P
C*ICA*O
h
OK
Cancel
Input
Play our game - guess the phrase
Enter one letter
CHICA*O
G
OK
Cancel



Grading

The programming questions are marked based on correctness, style of program and documentations. Add the external documentations in **documents.pdf**.

Submission

Submit a zip file (**Firstname-Surname-StudentId.zip)** including **documents.pdf, SecretPhrase.java and ArraysExercise.java** on Brightspace.