AUB American University of Beirut

الجامعة الأميركية في بيروت

Faculty of Arts & Sciences
Department of Computer Science
CMPS 200—Introduction to Programming
Exam 2

Problem 1 Finding Sets (40%)

An array of integers that represents a list of integers may contain repeated values. By contrast, an array of integers that represents a *set* does not contain any repetitions. Your task is to write a program `SetSize` that works on a given (hardcoded) array and outputs the `set` which corresponds to the given array. **The given array might be sorted or not sorted.**

- If the original array is sorted, the method `findSortedSet` should be called. `findSortedSet` accepts an `int` array as a parameter and returns another array representing the resulting *set*. After calling this method, output the contents of the original array, the contents of the resulting array (the *set*), and the number of elements removed.
- If the original array is NOT sorted, the method `findUnSortedSet` should be called. `findUnSortedSet` is very similar to `findSortedSet`(i.e. it accepts an `int` array and returns an `int` array), but it should operate on an unsorted array.

NOTE: You are not allowed to use `Arrays.sort` in this problem. Make your program as modular as possible

Sample Run 1 (Assume the input array is {0, 1, 2, 3, 4, 5, 6, 7, 8, 9})

```
C:> java SetSize
Array (Sorted) = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
Set = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
Elements Removed = 0;

C:>
```

Sample Run 2 (Assume the input array is {1, 5, 2, 8, 2, 9, 1, 0, -3, 5})

```
C:> java SetSize
Array (NOT Sorted) = {1, 5, 2, 8, 2, 9, 1, 0, -3, 5})
Set = {1, 5, 2, 8, 9, 0, -3})
Elements Removed = 3;

C:>
```

Sample Run 3 (Assume the input array is {1, 1, 1, 1, 1, 1, 1, 1, 1, 1})

```
C:> java SetSize
Array (Sorted) = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
Set = {1}
Elements Removed = 9;

C:>
```

Problem 2 HangMan (60%)

Hangman is a paper and pencil guessing game for two or more players. One player thinks of a word, phrase or sentence and the other tries to guess it by suggesting letters. In this program, `HangMan`, you need to simulate this game where you play against the computer. The process goes as follows:

1. The computer starts by picking a string. The computer can do so in two ways:
   a. Pick a string randomly from the following array:
      `String[] dictionary = {"I", "love", "this", "course", "tremendously", "it is the best course", "ever" };`
   b. Generate a random String. The generated string must not exceed the length of 10, is composed of only letters, and can include only one space (zero or one space).

   c. The computer decides on which way to pick a string by first generating a random number between 10 and 100, and checking whether the number is prime or not. If the number is not prime, the first method is adopted, otherwise, the second is adopted.
2. The user will then try to guess that string. Initially, the string is presented to the user as asterisks. The user is then asked to guess a letter. If the guess is correct, the letter(s) are revealed. The unknown letters will still be represented as asterisks. The process continues until the user knows all the letters in the string.
3. When the user finishes a word, display the number of misses and ask the user whether to continue for another word.
4. If the answer is yes, you should repeat the process. Note that strings that were picked in previous rounds, should not be picked again.

Make your program as modular as possible

Sample Run 1 (the user input is highlighted in bold below)

```
C:\> java HangMan

(Guess) Enter a letter in word ******* > p
(Guess) Enter a letter in word p****** > r
(Guess) Enter a letter in word pr**r** > p
p is already in the word
(Guess) Enter a letter in word pr**r** > o
(Guess) Enter a letter in word pro*r** > g
(Guess) Enter a letter in word progr** > n
n is not in the word
(Guess) Enter a letter in word progr** > m
(Guess) Enter a letter in word progr*m > a
The word is program. You missed 1 time

Do you want to guess for another word? Enter y or n> n
```