4 kyu Twice linear

Java:

```java
import java.util.TreeSet;
class DoubleLinear {

    public static int dblLinear (int n) {
      TreeSet<Integer> treeSet = new TreeSet<>();
      treeSet.add(1);

      for (int i = 0; i < n; i++) {
        int x = treeSet.pollFirst();
        treeSet.add(2 * x + 1);
        treeSet.add(3 * x + 1);
      }
      return treeSet.pollFirst();
    }
}
```

6 kyu 80's Kids #7: She's a Small Wonder

Java:

```java
import java.util.HashSet;
public class Robot {
  private HashSet<String> knownWords = new HashSet<>();
  public String learnWord(String word){
    knownWords.add("thank");
    knownWords.add("you");
    knownWords.add("for");
    knownWords.add("teaching");
    knownWords.add("me");
    knownWords.add("i");
    knownWords.add("do");
    knownWords.add("not");
    knownWords.add("understand");
    knownWords.add("the");
    knownWords.add("input");
    knownWords.add("already");
    knownWords.add("know");
    knownWords.add("word");
   if (knownWords.contains(word.toLowerCase()))
     return "I already know the word "+word;

   else if (!(word.toLowerCase().matches("[a-zA-Z]+")))
     return "I do not understand the input";

   else knownWords.add(word.toLowerCase());
   return "Thank you for teaching me "+word;
  }
 }
```

6 kyu Detect Pangram

Java:

```java
import java.util.HashSet;
public class PangramChecker {
  public boolean check(String sentence){
    HashSet<Character> hashSet = new HashSet<Character>();
    String str = sentence.toLowerCase().replaceAll("[^a-z]", "");
    for(int i = 0; i<str.length(); i++){
      hashSet.add(str.charAt(i));
    }
    if(hashSet.size()==26){
      return true;
    }else{
      return false;
    }
  }
}
```

6 kyu Your order, please

Java:

```java
import java.util.ArrayList;
public class Order {
  public static String order(String words) {
    String wordsInOrder ="";
    if(words.equals(" "))
      return " ";
    String splitWords[] = words.split(" ");
    for(int x = 1; x<=9; x++){
      for(int y = 0; y<splitWords.length; y++){
        if(splitWords[y].contains(String.valueOf(x)) && x != splitWords.length){
          wordsInOrder+=splitWords[y]+ " ";
        }
        if(splitWords[y].contains(String.valueOf(x)) && x==splitWords.length){
            wordsInOrder+=splitWords[y];
        }
      }
    }

    return wordsInOrder;

  }
}
```

6 kyu Which are in?

Java:

```java
import java.util.HashSet;
import java.util.TreeSet;
public class WhichAreIn {
    public static String[] inArray(String[] array1, String[] array2) {
    TreeSet<String> substring = new TreeSet<>();
    for(int i = 0; i<array1.length; i++){
      for(int y = 0; y<array2.length; y++){
        if(array2[y].contains(array1[i])){
          substring.add(array1[i]);
        }
      }
    }
    String subWords[] = new String[substring.size()];
    return substring.toArray(subWords);
    }
}
```