NAMA: ANDI DWI SAPUTRO

Hackerrank username: andiads06

A. Soal Medium

1. **Java Regex**

   Link: https://www.hackerrank.com/challenges/java-regex/problem

   Q: Write a class called *MyRegex* which will contain a string pattern. You need to write a regular expression and assign it to the pattern such that it can be used to validate an IP address.

   Accepted Answer:

Submitted Code

Language: Java 7 — Open in editor

```
19 class MyRegex{
20
21    String pattern0255 = "(\\d{1,2}|(0|1)\\d{2}|2[0-4]\\d|25[0-5])";
22
23     public String pattern = pattern0255 + "\\." + pattern0255 + "\\." + pattern0255 + "\\." + pattern0255;
24
25     public static void main(String[]args) {
26         System.out.println("000.12.12.034".matches(new MyRegex().pattern));
27         System.out.println("121.234.12.12".matches(new MyRegex().pattern));
28         System.out.println("23.45.12.56".matches(new MyRegex().pattern));
29         System.out.println("00.12.123.123123.123".matches(new MyRegex().pattern));
30         System.out.println("123.23".matches(new MyRegex().pattern));
31         System.out.println("Hello.IP".matches(new MyRegex().pattern));
32     }
33 }
```

B. Soal Easy

1. **Java Substring**

   Link: https://www.hackerrank.com/challenges/java-substring/problem

   Q: Print the substring in the inclusive range from start to end-1 .

   Accepted Answer:

Submitted Code

Language: Java 7 — Open in editor

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String S = in.next();
        int start = in.nextInt();
        int end = in.nextInt();
        System.out.println(S.substring(start, end));
    }
}
```

2. **Java Substring Comparison**

   Link: https://www.hackerrank.com/challenges/java-string-compare/problem

   Q: Return the respective lexicographically smallest and largest substrings as a single newline-separated string.

   Accepted Answer:

```java
public static String getSmallestAndLargest(String s, int k) {
    String smallest = "";
    String largest = "";

    smallest = s.substring(0,k);
    largest = s.substring(0,k);
    // Complete the function
    // 'smallest' must be the lexicographically smallest substring of length 'k'
    // 'largest' must be the lexicographically largest substring of length 'k'

    for (int i=0;i+k<=s.length();i++){
        if (s.substring(i,i+k).compareTo(smallest)<0){
            smallest = s.substring(i,i+k);
        }
        if (s.substring(i,i+k).compareTo(largest)>0){
            largest = s.substring(i,i+k);
        }
    }


    return smallest + "\n" + largest;
}
```

3. **Java String Reverse**

Link: https://www.hackerrank.com/challenges/java-string-reverse/problem

Q: Given a string , A print Yes if it is a palindrome, print No otherwise.

Accepted Answer:

Submitted Code

Language: Java 7 — Open in editor

```
6     public static void main(String[] args) {
7
8         Scanner sc=new Scanner(System.in);
9         String A=sc.next();
10        /* Enter your code here. Print output to STDOUT. */
11        String reversed = "";
12        char[] ch = A.toCharArray();
13        for (int i=ch.length-1;i>=0;i--){
14            reversed += ch[i];
15        }
16        if (A.equals(reversed)){
17            System.out.println("Yes");
18        } else {
19            System.out.println("No");
20        }
21
```

4. **Java Anagrams**

Link: https://www.hackerrank.com/challenges/java-anagrams/problem

Q: Print "Anagrams" if A and B are case-insensitive anagrams of each other; otherwise, print "Not Anagrams" instead.

Accepted Answer:

```
5     static boolean isAnagram(String a, String b) {
6         // Complete the function
7         a=a.toUpperCase();
8         b=b.toUpperCase();
9         boolean ret = false;
10        StringBuilder c= new StringBuilder(b);
11
12        if(a.length()==b.length()){
13            for(int i=0; i<a.length();i++){
14                for(int j=0; j<c.length();j++){
15                    if(a.charAt(i)==c.charAt(j)){
16                        c.deleteCharAt(j);
17                    if(i==a.length()-1 && c.length()==0){
18                        ret=true;
19                        break;
20                    }
21                    break;
22                }
23            }
24        }
25        }
26        return ret;
27    }
```

5. **Java String Tokens**

Link: https://www.hackerrank.com/challenges/java-string-tokens/problem

Q: Given a string s, matching the regular expression [A-Za-z !,?._'@]+, split the string into *tokens*. We define a token to be one or more consecutive English alphabetic letters. Then, print the number of tokens, followed by each token on a new line.

Accepted Answer:

Submitted Code

Language: Java 7 — Open in editor

```
9         // Write your code here.
10        s = s.trim();
11        String delimiters =  "[ !,?._'@]+";
12        String[] tokens = s.split(delimiters);
13        int n = tokens.length;
14        if (s.length()>400000){
15            System.out.println("");
16        } else if (s.length()==0){
17            System.out.println("0");
18        } else {
19            System.out.println(n);
20            for (int i=0;i<tokens.length;i++){
21                System.out.println(tokens[i].toString());
22            }
23        }
24
```

6. **Java Inheritance II**

Link: https://www.hackerrank.com/challenges/java-inheritance-2/problem

Q: Write the following code in your editor below:

1. A class named *Arithmetic* with a method named *add* that takes 2 integers as parameters and returns an integer denoting their sum.
2. A class named *Adder* that inherits from a superclass named *Arithmetic*.

Accepted Answer:

Submitted Code

Language: Java 7

```
//Write your code here
class Arithmetic {
    int add(int x, int y) {
        int sum = x+y;
        return sum;
    }
}

class Adder extends Arithmetic {
    int add(int x, int y){
        return super.add(x,y);
    }
}
```

7. **Java Interface**

Link: https://www.hackerrank.com/challenges/java-interface/problem

Q: You are given an interface *AdvancedArithmetic* which contains a method signature *int divisor_sum(int n)*. You need to write a class called MyCalculator which implements the interface.

Accepted Answer:

Submitted Code

Language: Java 7

```
7  //Write your code here
8  class MyCalculator implements AdvancedArithmetic {
9      int result = 0;
10     public int divisor_sum(int n){
11
12         for (int i=1;i<=n;i++){
13
14             if (n%i==0){
15                 result+=i;
16             }
17
18         }
19         return result;
20     }
21 }
```

8. **Java Method Overriding 2 (Super Keyword)**

Link: https://www.hackerrank.com/challenges/java-method-overriding-2-super-keyword/problem

Accepted Answer:

Submitted Code

Language: Java 7

```
18
19
20         String temp=super.define_me(); //Fix this line
21
22
```