IIP

Test Unit 6 - Possible solution

Year 2016-2017

Name:

1. (4 points) A possible approximation to the natural logarithm of a value $z \in \left[\frac{1}{2}, \frac{3}{2}\right]$ can be based on the development of the Taylor series that gives the following results:

$$\log(z) = \log(1+x) = \sum_{n=1}^{\infty} t_n \qquad t_1 = x \qquad t_n = (-1) \cdot t_{n-1} \frac{x \cdot (n-1)}{n} \quad n > 1$$

Implement a Java class (`static`) method that receives a value z (you can suppose that in the correct range) and a ϵ value that determines the allowed error and returns the approximation of the natural logarithm based on that formula. The stop criteria is when the absolute value of the last calculated term is lower than ϵ.

In the original version done in the classroom I forgot the (-1) in the recurrence, and it is not evaluated in the solution. Sorry for the inconveniences.

```
public static double log(double z, double eps) {
  double x = z - 1;
  double res = 0;
  int n = 1;
  double t = x;

  while (Math.abs(t) >= eps) {
    res+=t;
    n++;
    t = (-1)*t*x*(n-1)/n;
  }

  return res;
}
```

2. (2 points) Implement a Java class (`static`) method that receives a `String` and returns another `String` with the commom characters in the complementary positions (i.e., 0 and `length-1`, 1 and `length-2`, and so on). For example, for `antenna` it would return `anena`, for `redivider` it would return `redivider`, for `abcd` it would return the empty `String`.

```
public static String complement(String s) {
  String r="";
  int i;

  for (i=0;i<s.length();i++) {
    if (s.charAt(i) == s.charAt(s.length()-i-1))
      r+=s.charAt(i);
  }

  return r;
}
```

3. (4 points) Implement a program class that reads for two positive integer n and m and prints on the screen the following figure, where $2n-1$ is the maximum number of asterisks and m is the number of times the basic pattern is repeated and the number of final asteriks.

```
  *
 ***
*****
```

```
  *
 ***
*****
  *
  *
```

The previous case is for $n = 3$ and $m = 2$.

```java
import java.util.*;

public class XmasTree {
  public static void main(String [] args) {
    int n, m, i, j, k;
    Scanner kbd=new Scanner(System.in).useLocale(Locale.US);

    System.out.print("n: "); n=kbd.nextInt();
    System.out.print("m: "); m=kbd.nextInt();

    for (i=0;i<m;i++) {     // Repeat m times
      for (j=0;j<n;j++) {   // Loop for basic figure
        for (k=0;k<n-j-1;k++) System.out.print(" ");
        for (k=0;k<2*j+1;k++) System.out.print("*");
        System.out.println();
      }
    }

    for (i=0;i<m;i++) {    // Loop for final asterisks
      for (j=0;j<n-1;j++) System.out.print(" ");
      System.out.println("*");
    }
  }
}
```