Functions Gone Wild 2019

This problem will be worth 16 points. Each successfully implemented function will earn you a point. There is the student tester and one test for each of the 10 functions for a total of 11 possible points. Additional bonus points will be awarded for correctly implementing functions 1-3, functions 4 - 6, functions 7 and 8, and functions 9 and 10. A final bonus point will be awarded for correctly implementing all 10 functions.

This problem will ask you to implement the following functions. One function is defined recursively. On some problems I will give you functions from the java.lang.Math class with a brief description, and sometimes you just need to implement the function. In all the following functions you do **NOT** need worry about domain issues. That is, you may assume that all test data will not cause any exceptions to be thrown. In total, there are ten different functions in this problem. All methods which return a `double` must return a value 'close enough' (less than some delta > 0) to the correct answer to be considered correct.

You should use:

• `Math.max(a,b)` for max{a, b}.	• `Math.min(a, b)` for min(a, b)
• `Math.abs(x)` for |x|.	• `Math.cos(a)` for cos(a)
• `Math.sqrt(x)` for $\sqrt{x}$	• `Math.sin(a)` for sin(a)
• `Math.pow(x, 1.0/n)` for $\sqrt[n]{x}$	• `Math.tan(a)` for tan(a)
• `Math.log(a)` for $\ln(a)$	• `Math.log10(a)` for $\log_{10}(a)$ or $\log(a)$
• `Math.ceil(a)` for $\lceil a \rceil$.	• `Math.floor(a)` for $\lfloor a \rfloor$.

note:

- Return type of `Math.abs`, `Math.max` and `Math.min` is the same as its argument(s).
- Return type of all other methods is double.
- $\lfloor x \rfloor$ is the largest (Closes to positive infinity) double value smaller than or equal to x. For example, $\lfloor 2.9 \rfloor = 2$ and $\lfloor -14.3 \rfloor = -15$.
- $\lceil x \rceil$ is the smallest (Closes to negative infinity) double value that is greater than or equal to x. For example, $\lceil 1.1 \rceil = 2$ and $\lceil -24.9 \rceil = -24$.

- Use the following constant for pi (π)

`Math.PI`	The `double` value that is closer than any other to *pi*, the ratio of the circumference of a circle to its diameter.

- Use the following constant for e

`Math.E`	The `double` value that is closer than any other to *e*, the base of the natural logarithms.

Functions Gone Wild 2019

1. Define the happyFace (☺) operator as: a ☺ b = a + ab + b

   Then define happyFace power as $a^{☺n}$ = (…((a ☺ a) ☺ a) ….. ☺ a) {a is written n times}
   with $a^{☺0} = 0$, and $a^{☺1} = 0☺a = a$

   Implement the happyFace power function.

Use following function heading:

```
public static int f1(int a, int n)
```

```
Test data:  f1(2, 1) returns 2 = 2^☺1
            f1(2, 2) returns 8 = 2^☺2 = (2+2*2+2)
            f1(2, 3) returns 26 = 2^☺3 = (2+2*2+2) ☺ 2
                                   = 8 ☺ 2 = 8 + 2*8 + 2
            f2(4, 2) returns 24 = 4^☺2 = (4+4*4+4)
            f3(3, 5) returns 1023 = 3^☺5 = ((((3+3*3+3) ☺ 3) ☺ 3) ☺ 3)
                                   = (((15 + 15*3 + 3) ☺ 3) ☺ 3
                                   = ((63 + 63*3 + 3) ☺ 3)
                                   = 255 ☺ 3 = 255 + 765 + 3
                                   = 1023
```

2. According to gmatclub.com: In the x-y plane, point (p, q) is a lattice point if both p and q are integers. Return the number of lattice points either on the border or in the interior of the rectangle bounded by the lines:
   $x = a1$
   $x = a2$
   $y = b1$
   $y = b2$

   You may assume $a1 < a2$ and $b1 < b2$

Use following function heading:

```
public static int f2(double a1, double a2, double b1, double b2)
```

```
Test data:  f2(0.5, 3.6, 1.1, 8.1) returns 21
            f2(-1.3, 5.0, -4.6, -0.9) returns 28
            f2(-3.1, 2.9, 0.95, 9.03) returns 54
```

Functions Gone Wild 2019

3. Implement the following integer recursive function. All calculations shall be computed using Integer math.

$$f3(n)=\begin{cases} f3\left(\frac{3n+2}{7}-1\right)-\frac{n}{10} & \text{n} \geq 100, \text{n is even} \\ f3\left(\frac{2n+11}{n+1}\right)+3n-7 & \text{n} \geq 100, \text{n is odd} \\ \frac{f3\left(\frac{3n}{11}\right)}{2}-2f3\left(\frac{n+18}{3}\right) & 25 < \text{n} < 100 \\ (n+3)^2-1 & \text{otherwise} \end{cases}$$

Use following function heading:

```
public static int f3(int n)
```

```
Test data:  f3(202) returns 1771
            f3(135) returns 422 = 398 + f(2) = 398 + 24
            f3(35) returns -727
            f3(5) returns 63
```

4. Implement the following function

Consider the mathematical notation $\sum_{i=m}^{n}(someFunction)$ used to represent the summation of many similar terms.

The notation $\sum_{i=m}^{n}(h(i))$ is defined as: $h(m)+h(m+1)+h(m+2)+...+h(n)$

The subscript gives the symbol for an index variable, i. Here, i represents the index of summation; m is the lower bound of summation, and n is the upper bound of summation. In this case, i = m under the summation symbol means that the index i starts <u>equal</u> to m. Successive values of i are found by adding 1 to the previous value of i, continuing up to and including when i <u>equals</u> n. An example: $\sum_{k=2}^{6}k^2=2^2+3^2+4^2+5^2+6^2=90$.

Your task in this problem is to implement the following function.

Note – all calculations are to be completed using integer math

$$f4(a,b,c)=\sum_{i=\min(ba-c,bc-a)}^{\max(a+bc,c+ab)}\left(i\frac{c+2bi}{|a+i|}+\frac{b(c+3i)}{1+a}+\left(\frac{abi}{c}\right)\right)=$$

Functions Gone Wild 2019

$$\min(ba-2c,bc-a)\frac{c+2b\min(ba-2c,bc-a)}{|a+\min(ba-2c,bc-a)|}+\frac{b(c+3\min(ba-2c,bc-a))}{1+a}+\frac{ab\min(ba-2c,bc-a)}{c}+$$

$$[1+\min(ba-2c,bc-a)]\frac{c+2b[1+\min(ba-2c,bc-a)]}{|a+[1+\min(ba-2c,bc-a)]|}+\frac{b(c+3[1+\min(ba-2c,bc-a)])}{1+a}$$

$$+\frac{ab[1+\min(ba-2c,bc-a)]}{c}+$$

$$+.....+$$

$$\max(a+bc,c+ab)\frac{c+2b\max(a+bc,c+ab)}{|a+\max(a+bc,c+ab)|}+\frac{b(c+3\max(a+bc,c+ab))}{1+a}+\frac{ab\max(a+bc,c+ab)}{c}$$

Special Note: $if\ \max(a+b*c,c+b*a)<\min(b*a-c,b*c-a)$ return 0.

Use following function heading:

```
public static int f4(int a, int b, intc)
```

```
Test data:
   f4(1, 1, 1)
        returns = f(0) + f(1) … + f(2) = 0 + 4 + 8 = 12


   f4(2, 2, 2)
        returns = f(2)+f(3)+f(4)+f(5)+f(6) =  14+21+29+36+44 = 144

   f4(2, 1, 2)
        returns = f(0)+f(1)+f(2)+f(3)+f(4) =  0+3+7+10+14 = 34
```

Functions Gone Wild 2019

5. Let's do a little chemistry ☺
   The pH of a solution is a measure of the ionization of water to form H^+ (acids) or OH^- (bases).

   There are four important variables (pH, pOH, H^+, and OH^-) and two important equations:

   $$pH = -\log_{10}(H^+)$$
   $$pOH = -\log_{10}(OH^-)$$

   Here the logs are logs base 10.The variables pH and pOH are also related by the equation:

   $$pH + pOH = 14$$

   Implement the following function which returns `pH`. Output a String representing the value rounded to two decimal places (the hundredths place value)

Use following function heading:

```
public static String f5(String concentration, double level)
```

```
Test data:  f5("H",0.001)   returns "3.00"
            f5("OH", 0.001) returns "11.00"
            f5("H", 0.012)  returns "1.92"
```

6. Implement the following function

$$f6(x,y,z)=\begin{cases}\log 10\left(\left|\pi^{3y-2e}\right|\right)+\ln\left(\left|(ze)^{x+e}\right|\right) & \ln\left(\left|x+z\right|\right)<y \\ \log 10\left(e\left|x-y\right|\right)+\left|\ln\left(\left|z-y\right|\right)\right|^{|xz|} & \ln\left(\left|x+z\right|\right)\geq y\end{cases}$$

Use following function heading:

```
public static double f6(double x, double y, double z)
```

```
Test data:  f6(2, 2, 3)  returns  10.1819565
            f6(2, 1, 3)  returns  0.5451999
```

Functions Gone Wild 2019

7. Implement the following function which scores how similar are two different `Strings`. The score is calculated by comparing corresponding 'letters' in the `Strings` according to the following:

   - If the corresponding letters are matching upper case letters, increase score by 5.
   - If the corresponding letters are matching lower case letters, increase score by 4.
   - If the corresponding letters match but are different case (Lower/Upper), increase score by 3
   - If the corresponding letters are matching non letters (only need to check for @, #, & or *) increase score by 2
   - All others do not increase or decrease score.

You may assume the String will contain only letters or the four non letters: @, #, & and *

Helpful methods in `String` class:
`public String toLowerCase()` Converts all of the characters in this String to lower case.
For example: `"ANd$Y".toLowerCase()` returns `"and$y"`

`public String toUpperCase()` Converts all of the characters in this String to upper case.
For example: `"Abc@Ws".toUpperCase()` returns `"ABC@WS"`

Use following function heading:

```
public static int f7(String phr1, String phr2)
```

```
Test data:  f7("CODE", "CODE") returns 20 (5 + 5 + 5 + 5)
            f7("SCore", "Score") returns 20 (5 + 3 + 4 + 4 + 4)
            f7("@Precede*", "#proceed*") returns 17  (0+3+4+0+4+4+0+0+2)
            f7("#abcde", "#acCex") returns 9 (2+4+0+3+0+0)
```

Functions Gone Wild 2019

8 Implement the following function

This method will scramble the `String` phrase by replacing each individual letter using a simple letter substitution scheme. The scramble is performed by replacing vowels in the sequence: (a, e, i, o, u, y) with the vowel advanced three letters, cyclically.

Similarly, consonants are replaced from the sequence (b k x z n h d c w g p v j q t s r l m f) by advancing ten letters, cyclically.

You may assume the String contains only lower case letters and spaces (spaces remain unchanged).

Use following function heading:

```
public static String f8(String phrase)
```

```
Test data:   f8("bcdfghjklmnpqrstvwxz") returns "plrgfsxvcwtbzdhnkmjq"
             f8("plrgfsxvcwtbzdhnkmjq") returns "bcdfghjklmnpqrstvwxz"

             f8("aeiouy") returns "ouyaei"
             f8("ouyaei") returns "aeiouy"

             f8("good job") returns "faar xap"
             f8("faar xap") returns "good job"
```

Functions Gone Wild 2019

9. Implement the following function

$$f9(x,y,z)=\begin{cases} false & x=false & y=false & z=false \\ true & x=false & y=false & z=true \\ false & x=false & y=true & z=false \\ false & x=false & y=true & z=true \\ true & x=true & y=false & z=false \\ true & x=true & y=false & z=true \\ false & x=true & y=true & z=false \\ false & x=true & y=true & z=true \end{cases}$$

Use following function heading:

```
public static boolean f9(boolean x, boolean y, boolean z)
```

```
Test data:  f9(false, false, false) returns true
```

10. Implement the following function

$$f10(j,k,m,n)=\begin{cases} false & j=false & k=false & m=false & n=false \\ true & j=false & k=false & m=false & n=true \\ false & j=false & k=false & m=true & n=false \\ false & j=false & k=false & m=true & n=true \\ true & j=false & k=true & m=false & n=false \\ true & j=false & k=true & m=false & n=true \\ false & j=false & k=true & m=true & n=false \\ true & j=false & k=true & m=true & n=true \\ false & j=true & k=false & m=false & n=false \\ false & j=true & k=false & m=false & n=true \\ false & j=true & k=false & m=true & n=false \\ true & j=true & k=false & m=true & n=true \\ false & j=true & k=true & m=false & n=false \\ true & j=true & k=true & m=false & n=true \\ false & j=true & k=true & m=true & n=false \\ true & j=true & k=true & m=true & n=true \end{cases}$$

Use following function heading:

```
public static boolean f10(boolean j, boolean k, boolean m, boolean n)
```

```
Test data:  f10(false, false, false, false) returns true
```