# Functions Gone Wild 2018

**<u>This problem will be worth 16 points</u>**. 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 7and 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 problem I will give you functions from the java.lang.Math with a brief description, and some times you just need to implement the function. In all the following functions you do **<u>NOT</u>** 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 value within 0.05 of 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 ( $\pi$ )

| | |
|---|---|
| `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 2018

1. Implement the following function with parameter `n` which returns the smallest positive (greater than 0) value of `d` such that `n + d` is a perfect square?

   According to Wikipedia: In mathematics, a perfect square is an integer that is the square of an integer; in other words, it is the product of some integer with itself. For example, 9 is a perfect square, since it can be written as $3 \times 3$.

   You may assume n > 0, and the return value must be greater than or equal to 0.

```
public static int f1(int n)

Test data:  f1(8) returns 1        // since 3^2 = 9 = 8 + 1
            f1(25) returns 11      // since 6^2 = 36 = 25 + 11
            f1(64) returns 0       // since 8^2 = 64 = 64 + 0
            f1(90) returns 10      // since 10^2 = 100 = 90 + 10
```

The correct call to the method is of the form: `FunctionsGoneWild2018.f1(n)`

- This problem borrowed from March edition of NCTM monthly magazine.

2. Implement the following function which returns the number of `int`s that satisfy the inequality:

   $low < 5y + 7 <= high$

   You may assume $low < high$ .

```
public static int f2(int low, int high)

Test data:  f2(0, 100) returns 20        //   the ints [-1, 18]
            f2(26, 52) returns 6         //   the ints [4, 9]
            f2(-30, 200) returns 46      //   the ints [-7, 38]
```

The correct call to the method is of the form: `FunctionsGoneWild2018.f2(low, high)`

- This problem borrowed from March edition of NCTM monthly magazine.

# Functions Gone Wild 2018

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

$$f3(n) = \begin{cases} f3\left(\frac{3n}{5} - 1\right) - 2 & n \geq 100, n \text{ is even} \\ f3\left(\frac{2n-13}{21}\right) - \frac{n}{3} & n \geq 100, n \text{ is odd} \\ f3\left(\frac{2n}{5}\right) - f3\left(\frac{n-5}{3}\right) / 2 & 50 < n < 100 \\ n^2 + 3n - 19 & \text{otherwise} \end{cases}$$

using the following function heading:

```
public static int f3(int n)

Test data:  f3(202) returns 580
            f3(135) returns 116
            f3(55) returns 389
            f3(35) returns 1311
```

The correct call to a recursive static method is of the form: `FunctionsGoneWild2018.f3(n)`

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: $h(m)+h(m+1)+h(m+2)+\ldots+h(n)$

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 to</u> m. Successive values of i are found by adding 1 to the previous value of i, stopping 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.

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

| Special note – all calculations are to be completed using integer math |
|---|

# Functions Gone Wild 2018

$$\left[\begin{array}{l} \min(b*a-c,b*c-a)\dfrac{c+2b\min(b*a-c,b*c-a)}{1+\left|a-\min(b*a-c,b*c-a)\right|} + \dfrac{b+(c+3)a\min(b*a-c,b*c-a)}{1+\dfrac{\min(b*a-c,b*c-a)}{c}} \\ \qquad +\left(\min(b*a-c,b*c-a)+3ab\right)\min(b*a-c,b*c-a) \end{array}\right]+$$

$$\left[\begin{array}{l} \left[1+\min(b*a-c,b*c-a)\right]\dfrac{c+2b\left[1+\min(b*a-c,b*c-a)\right]}{1+\left|a-\left[1+\min(b*a-c,b*c-a)\right]\right|} + \dfrac{b+(c+3)a\left[1+\min(b*a-c,b*c-a)\right]}{1+\dfrac{\left[1+\min(b*a-c,b*c-a)\right]}{c}} \\ \qquad +\left(\left[1+\min(b*a-c,b*c-a)\right]+3ab\right)\left[1+\min(b*a-c,b*c-a)\right] \end{array}\right]+$$

$$+\ldots+$$

$$\left[\begin{array}{l} \max(a+b*c,c+a*b)\dfrac{c+2b\max(a+b*c,c+a*b)}{1+\left|a-\max(a+b*c,c+a*b)\right|} + \dfrac{b+(c+3)a\max(a+b*c,c+a*b)}{1+\dfrac{\max(a+b*c,c+a*b)}{c}} \\ \qquad +\left(\max(a+b*c,c+a*b)+3ab\right)\max(a+b*c,c+a*b) \end{array}\right]$$

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

```
Test data:
   f6(2, 3, 4)
        returns = f(2) + f(3) … + f(14) =
                     103+141+154+193+235+278+305+351+398+448+484+537+591
                     = 4218
   f6(1, 9, 1)
        returns = f(8) + f(9) + f(10) =  429 + 491 + 555 = 1475
```

The correct call to the static method is of the form: `FunctionsGoneWild2018.f4(a, b, c)`

# Functions Gone Wild 2018

5. Implement the following function

$$f5(x, y) = \left| \frac{\tan(e - 3y)}{\sin(y)} \right|^{\pi \cos(x)}$$

Use following function heading:

```
public static double f5(double x, double y)
```

```
Test data:  f5(1, 3)  returns 0.0004303
            f5(2, -1) returns 1.448499
```

The correct call to the static method is of the form: `FunctionsGoneWild2018.f5(x, y)`

6. Implement the following piecewise function.

$$f6(x, y, z) = \begin{cases} \pi^{x/e} + \left[\log(|y+z|)\right]^{\log(|x+y|)} & \ln(|x|) > (y+z) \\ \log(|x-y|) + \ln(|z-y|) & \ln(|x|) \le (y+z) \end{cases}$$

Use following function heading:

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

Test data: `f6`$\left(-e^{3}, -1, 3\right)$ `returns 0.204239 …`
`f6(25, 2, 1)  returns  1.36172 …`

The correct call to the static method is of the form: `FunctionsGoneWild2018.f6(x, y)`

# Functions Gone Wild 2018

7. Implement the following function

ok – The fun in this problem is using the following Test data to figure it out..

```
public static String[] f7(String phrase, int num)
```

Special Note: `phrase.length() > 0 and num > 0`.

```
Test data:
  f7(“SAMPLE”, 2)
          returns new String[] {“SML”, “APE”}

  f7(“HELP ME FIGURE THIS OUT :(”, 3)
          returns new String[] {“HPEIRTSU:”, “E  GEH T(”, “LMFU IO ”}

  f7(“COMPUTER SCIENCE”, 4)
          returns new String[] {“CU E”, “OTSN”, “MECC”, “PRIE”}

  f7(“TO ITERATE IS HUMAN, TO RECURSE DIVINE”, 5)
          returns new String[] {“TE U EEI”, “ORIMTC N”,
                                          “ ASAOUDE”, “IT N RI”, “TEH,RSV”}

  f7(“1234567890”, 5)
          returns new String[] {“16”, “27”, “38”, “49”, “50”}

  f7(“@ABC!”, 7)
          returns new String[] {“@”, “A”, “B”, “C”, “!”, “”, “”}
```

The correct call to the static method is of the form: `FunctionsGoneWild2018.f7(phrase, num)`

# Functions Gone Wild 2018

8. Implement the following function


Consider the letter only keyboard shown to the right. This method will scramble the `String` phrase by replacing each individual letter with the letter on the opposite side of the keyboard. You only need to worry about Capital letters (and spaces are left unchanged).

The last test data sample scrambles the String:
`"THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG"`.
It should be noted this string contains every letter in the alphabet.

Use following function heading:

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

Special Note: If `phrase.length() == 0`, return a String of length zero.

```
F8 Test data
Test data:
    F8("COMPUTER SCIENCE")
            returns "BWZQRYIU KBEIXBI"

    F8("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG")
            returns "YFI PREBS CUWOX HWN DRZQK WVIU YFI ALMT JWG"
```

The correct call to the static method is of the form: `FunctionsGoneWild2018.f8(phrase)`

# Functions Gone Wild 2018

9. Implement the following function

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

Use following function heading:

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

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

The correct call to the static method is of the form: `FunctionsGoneWild2018.f9(x, y, z)`

10. Implement the following function

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

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
```

The correct call to the static method is of the form: `FunctionsGoneWild2018.f9(j, k, m, n)`