Specifications

Problem Description

You are going to write a program that calculates a mouse's time through a maze based on mouse characteristics.

Input: You will obtain the mouse name, category, and points for five characteristics (curiosity, hunger, sense of smell, caution, and speed) from the user. You may assume the user enters the correct type of data at each prompt, but you may not assume that the user enters a valid mouse category or numeric data in the valid range. Your program must validate the inputs.

Output: If all of the input data is valid, the program will compute the mouse's time through the maze according to the formulae in the design specifications section, displayed to two decimal places.

If any input data is invalid, your program will display error messages for each invalid data item after all input has been entered and will not calculate a time.

Interface: This is a console program. Do not use JOptionPane. Your program should prompt the user for data in the following order: points for curiosity, hunger, smell, caution, and speed, then name, category. Your program should use the following greeting:

```
You have fifteen points to distribute among five mouse qualities:
      Curiosity
      Hunger
      Sense of smell
      Caution
      Speed
Each quality must have at least one point.
```

Your program should use the following prompts:

- `How many points for curiosity?`
- `How many points for hunger?`
- `How many points for sense of smell?`
- `How many points for caution?`
- `How many points for speed?`
- `Enter a name for your mouse:`
- `Enter A, B, or C for mouse type:`

The program will validate the inputs as follows:

- Ensure that the characteristics points do not add up to more than 15. (Less than 15 is okay.)
- Ensure that a value of at least 1 has been assigned to each characteristic.
- Ensure that the mouse type is either "A", "B", or "C", either uppercase or lowercase ("a", "b", "c" are okay for inputs).

Use the following error messages (exactly) for invalid input (substituting user-entered values for underlined 16, and D):

- `You've used 16 points – only 15 are allowed.`
- `Each mouse quality must be assigned a minimum value of 1.`
- `D is not a valid mouse type.  Mouse type must be A, B, or C.`

If all data is valid, the program will print a message using the following text (substitute user-entered and calculated result for underlined mouse's name and time):

`Mortimer's time through the maze is 3.48 seconds.`

Design specifications: Use only techniques that are covered in the first two lessons. Use console input and output.

Your program must use one or more `if` statements, `switch` statements, and `Boolean` variables. Choose the best use for each.

Do not, under any circumstances, use `System.exit` or `return`. Use `break` only in `switch` statements. Do not submit code that includes these constructs (except for `break` in `switch` statements). Submit whatever you can write that does not include these constructs.

The software will first obtain all inputs. It will then validate the inputs according to specifications in the interface section.

If (and only if) all inputs are valid, the program will compute the mouse's time through the maze according to these formulae:

Mouse type A:

$$time = \ln(curiosity) + \log_2(speed) + \ln(smell)$$

Mouse type B:

$$time = \ln(caution) + \log_2(hunger) + \ln(smell)$$

Mouse type C:

$$time = \ln(hunger) + \log_2(speed) + \ln(caution)$$

In order to compute a natural log (ln), use Math.log(x). In order to compute a log base 2, use Math.log(x) / Math.log(2). So for example, with variables named `curiosity`, `speed`, and `smell`, the time computation for mouse type A would be:

```
time = Math.log(curiosity) + Math.log(speed) / Math.log(2) + Math.log(smell);
```

Results should be displayed to two decimal places.

Testing

Test Case 1

Purpose
Test a case with all valid inputs.

Input
Curiosity: 5
Hunger: 2
Smell: 3
Caution: 4
Speed: 1
Name: Mortimer
Type: B

Expected output
Results were calculated using the Windows scientific calculator.

```
Mortimer's time through the maze is 3.48 seconds.
```

Test Case 2

Purpose
Test a case with all invalid inputs.

Input
Curiosity: 0
Hunger: 0
Smell: 0
Caution: 0
Speed: 16
Name: Minnie
Type: D

Expected output
```
Each mouse quality must be assigned a minimum value of 1.
You've used 16 points – only 15 are allowed.
D is not a valid mouse type.  Mouse type must be A, B, or C.
```

Rubric
An exceptional-quality assignment will meet the following standards:

- Meeting functional and design specifications
  The Java program works and meets all of the specifications, with no additional unspecified functionality. The programmer has used programming techniques from the first and second lessons only. If the program misses specifications or does not function correctly, errors are acknowledged with a thorough and reflective analysis in the testing section (points will be removed for missed specifications).
- Communicating with identifiers and white space
  The program makes appropriate use of variables. Variables and constants are named according

to convention and are named for understandability and purpose. White space, both vertical and horizontal, is correctly used for readability and meets programming conventions.

- Communicating through documentation
  The Java program contains comments including the programmer's name and date. There are block comments (as many as necessary) for each distinct block of code which accurately describe what the block is accomplishing by relating the code to the problem being solved. Javadoc is included and meets the javadoc standards.
- Assumptions and Testing
  Testing is thorough. If there are errors, they are described in the testing section. If there are questions, they are answered thoughtfully in the testing section. All assumptions are made explicit.