

Problem Set 2: Fundamental Objects

[3 pts] You may only use the built-in mathematical functions, operators, functions that were introduced in class up to the fifth lecture, and additional functions that are listed below. Name the functions and their formal arguments exactly as shown. No paper submission is required for this assignment. Upload all R code onto Canvas, with a separate file for each problem; do not include any code for testing your functions. Name your files `Group \mathfrak{P} \mathfrak{Q} .R`, where \mathfrak{P} is the letter assigned to your group and \mathfrak{Q} is the problem number. List the names of the contributors at the top of each file, and make sure that the files can be sourced without error or it will not be graded.

The following functions, which have not been introduced in class, may or may not be useful for this problem set:

```
cat, is.character, is.complex, is.integer, is.logical,
is.numeric, is.raw, sprintf
```

Read the help files to learn about these functions.

1. **[10 pts]** Recall that a quadratic equation, $ax^2 + bx + c = 0$, can be solved using the formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

The formula will produce imaginary roots if the discriminant, $b^2 - 4ac$, is less than zero, one real root if the discriminant equals zero, and two real roots if the discriminant is positive. Suppose that we are not interested in imaginary roots. Write an R function named `solveQuadr` that:

- accepts the three coefficients, a , b and c , as its arguments (do not use a vector argument for this exercise), with a default value of 1 for a ;
- prints the following when the roots are imaginary:

```
> solveQuadr(1, -3, 4)
There are no real roots.
```
- prints *exactly one* root when there is only one root:

```
> solveQuadr(-4, 12, -9)
root: 1.5
```
- prints both roots when there are two real roots:

```
> solveQuadr(2, -1, -8)
1st root: -1.765564
2nd root:  2.265564
```
- returns (but do not print) the value 0 if there are no real roots and 1 otherwise.

2. [10 pts] Write an R function named `getDigit` that accepts a real number `x` and an integer `n` as its arguments, and returns the digit in the `n`th decimal place of `x`. It should signal an error, using the function `stop`, if it is called with one or more invalid arguments. Some examples of calling the function include:

```
> getDigit(1.234, n = 2)
[1] 3
> getDigit(1.234, n = -1)
Error in getDigit(1.234, n = -1) : n has to be a positive integer.
> getDigit(1, n = 2)
[1] 0
```
