

MIS2402 – Hands On Prep

Part A – Multiply the natural numbers between 1 and x.

- 1) Download prep.zip to your computer.
- 2) Working in prepa.html found in the prep folder, complete the function getFactorial.
 - a. You should assume that the user has entered an integer. Don't worry about checking for non-numeric user input, or for user input with decimal precision.
 - b. If the user entered a negative number, the function should return the string "error"
 - c. If the user entered a zero, then the function should return the number: 1
 - d. If the user enters a number of 1 or higher, return the product of the numbers between 1 and x. For example, if the user entered 4, the function should return 24 because $1 \times 2 \times 3 \times 4$ equals 24.
 - e. HINT: You will need a for loop to solve this problem.
- 3) Check your work.
- 4) Fun fact: In mathematics, the product of all the natural numbers between 1 and x is also called "x factorial". That's why the function is called getFactorial.

Part B – Using loops to generate text

- 5) Working in prepb.html found in the prep folder, complete the solution as instructed:
 - a. In this part, we **won't assume** that the user has entered an integer.
 - b. This time, we will put our error trap for non-natural user input into the **event handler**.
 - c. If the user gave you bad input, put "Bad data. Try again." into the div tag textDisplayed1
 - d. If the user gave your good input, call the makeStaircase function and put the output of that function into textDisplayed1
 - e. Complete the function makeStaircase. It should produce a string of "X" characters as and "
" tags as shown in the illustrations below.
 - f. HINT: To solve this problem, you will need a for loop inside another for loop.

Staircase

Enter an integer x.

Your number x:

X

Staircase

Enter an integer x.

Your number x:

X
XX
XXX

CONTINUED

Staircase

Enter an integer x.

Your number x:

X
XX
XXX
XXXX
XXXXX

Staircase

Enter an integer x.

Your number x:

Bad data. Try again.

- 6) Check your work
- 7) In this example, we see that error trapping may occur in the event handler. You have also seen error trapping in the function itself. *Always read all quiz and assignment instructions carefully to determine where the error trapping is expected, and where it is not!*
- 8) In this example, we also saw that a loop counter (i) does not always need to be part of a mathematical formula. Sometimes, it just controls the loop.