Numbers

1. Write a function which accepts two numbers and returns the sum.
   Ex: f[2;3] returns 5

2. Write a function which accepts two numbers and prints out the result as "[num1] + [num2] = [result]" on its own line.
   Ex: f[2;3] prints 2 + 3 = 5 and the cursor ends up on the next line

3. Write a function that sorts an input integer list in descending order and returns the sorted list.
   Ex: f[3 2 4 1 5] returns 5 4 3 2 1

4. Write a function that returns all non-negative integers smaller than an input integer.
   Ex: f[5] returns 0 1 2 3 4

5. Write a function that finds the ceiling of a float input.
   Ex: f[3.1] returns 4, f[-3.1] returns -3

6. Write a function that, given a list of lists of numbers, returns the first number of each list.
   Ex: f[(1 2 3; 4 5; 6 7 8 9)] returns 1 4 6

7. Write a function that, given a list of lists of numbers, returns the size of each list.
   Ex: f[(1 2 3; 4 5; 6 7 8 9)] returns 3 2 4

8. Write a function that multiplies the corresponding elements of two input integer vectors of the same size.
   Ex: f[1 2 3; 4 5 6] returns 4 10 18

9. Write a function that accepts two integer vectors and adds the entire second vector to each element of the first vector, returning the result.
   Ex: f[1 2 3; 4 5] returns (5 6; 6 7; 7 8)

10. Write a function that accepts two integer vectors and adds the entire first vector to each element of the second vector, returning the result.
    Ex: f[1 2 3; 4 5] returns (5 6 7; 6 7 8)

11. Write out what 1 2 3 +/:\: 4 5 and 1 2 3 +\:/: 4 5 are doing, step by step.

12. Write a function that sums up the elements of an input integer vector and returns the result.
    Ex: f[1 2 3 4] returns 10

13. Write a function that returns the sum of the first n numbers where n is the only parameter to the function. First try it using a loop. Then try it without loops.
    Ex: f[5] returns 15

14. Write a function that returns the sum of all the even numbers up until n, where n is the only parameter to the function.
Ex: f[5] returns 6

Strings

1. Write a function which accepts a string str and a non-negative integer n and returns the first n characters of str.
   Ex: f["abcde"; 3] returns "abc"

2. Write a function which accepts a string str and a non-negative integer n and returns the last n characters of str.
   Ex: f["abcde"; 3] returns "cde"

3. Write a function which accepts a string str, a non-negative integer beg, and a non-negative integer end and returns the substring starting at index beg and ending at index end.
   Ex: f["abcde"; 1; 3] returns "bcd"

4. Write a function which accepts a string str and returns that string with its letters sorted in ascending order.
   Ex: f["adceb"] returns "abcde"

5. Write a function which accepts a string str and returns the unique characters in that string.
   Ex: f["aecabcab "] returns "aecb"

6. Write a function which accepts a string str and returns a 2-element list where the first element is the list of the unique characters in that string and the second element is a list containing the count of each unique element in the string.
   Ex: f["aecabcab"] would return ("aecb"; 3 1 2 2)

7. Write the same function as above but with the return type being a dictionary where the keys are the unique characters of the string and the values are the counts. Hint – try it by modifying your previous answer with a few characters.
   Ex: f["aecabcab"] would return .((`a;3); (`e;1); (`c;2); (`b;2))

8. Write a function which accepts a list of strings and appends ".txt" to the end of each of them, returning the modified list.
   Ex: f[("file1";"file2";"file3")] returns ("file1.txt";"file2.txt";"file3.txt")

9. Write a function that accepts a symbol sym and a string str which concatenates the symbol, a dot, the string, and returns the result as a symbol
   Ex: f[`abc;"def"] returns `abc.def

10. What are the dangers of the function {[paths] `4: "rmdir /s /q " ,/: paths}? Note: Use `0: to play around with inputs instead of `4: so you can see what would have been passed to `4: without

actually executing it.

11. Write a function that accepts a list of integers/floats/symbols/characters or strings and a delimiter character (i.e. ",") and creates a single delimited string out of the elements in that list.
    Ex: f[1 2 3 4;"|"] returns "1|2|3|4"

12. Write a function that accepts a list of symbols and a string which appends the string to the symbol list (i.e. the symbol list grows by one element where that element is the string).
    Ex: f[`abc `def; "ghi"] returns (`abc;`def; "ghi")

13. Write a function that trims leading and trailing spaces in the input string, returning the resulting string.
    Ex: f["   ab cd   e   "] returns "ab cd   e"

14. Write a function that consolidates all consecutive spaces of an input string into one and returns the modified string.
    Ex: f["   ab cd   e   "] returns " ab cd e "

15. Write a function which accepts a string and a delimiter and breaks that string up into tokens using the given delimiter. Advanced: Make it ignore delimiters that appear within quotes.
    Ex: f["abc,def,ghi"; ","] returns ("abc"; "def"; "ghi")
        f["abc,\"def,ghi\""; ","] returns ("abc";"\"def,ghi\"")

16. Write a function that returns the current local time as a string in "YYYY/MM/DD HH:MM:SS" format

17. Write a function that returns a list of all the absolute paths of files within a given directory on the file system, including files within subdirectories. If the input is a file path, returns a list containing only that file path.