Problem 1. (Palindrome) Implement the function `isPalindrome()` in `palindrome.py` that returns `True` if the argument `s` is a palindrome (i.e., reads the same forwards and backwards), and `False` otherwise. You may assume that `s` is all lower case and doesn’t contain any whitespace characters.

```
~ /workspace/exercise4
$ python3 palindrome.py bolton
False
$ python3 palindrome.py amanaplanacanalpanama
True
```

Problem 2. (Sine Function) Implement the function `sin()` in `sin.py` that calculates the sine of the argument `x` in radians, using the formula

$$
\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots.
$$

Note: to test for convergence, use the condition similar to the one in the `cdf()` function from the `gaussian.py` library we discussed in class.

```
~ /workspace/exercise4
$ python3 sin.py 60
0.8660254037844385
```

Problem 3. (Euclidean Distance) Implement the function `distance()` in `distance.py` that returns the Euclidean distance between the vectors `x` and `y` represented as one-dimensional lists of floats. The Euclidean distance is calculated as the square root of the sums of the squares of the differences between the corresponding entries. You may assume that `x` and `y` have the same length.

```
~ /workspace/exercise4
$ python3 distance.py 5
-9 1 10 -1 1
5
-5 9 6 7 4
13.0
```

Problem 4. (Reverse) Implement the function `reverse()` in `reverse.py` that reverses the one-dimensional list `a` in place, i.e., without creating a new list.

```
~ /workspace/exercise4
$ python3 reverse.py
to be or not to be that is the question
<ctrl-d>
```

Problem 5. (Transpose) Implement the function `transpose()` in `transpose.py` that creates and returns a new matrix that is the transpose of the matrix represented by the argument `a`. Note that `a` need not have the same number rows and columns. Recall that the transpose of an $m$-by-$n$ matrix $A$ is an $n$-by-$m$ matrix $B$ such that $B_{ij} = A_{ji}$, where $0 \leq i < n$ and $0 \leq j < m$.

```
~ /workspace/exercise4
$ python3 transpose.py
2 3
1 2 3
4 5 6
1.0 4.0
2.0 5.0
3.0 6.0
```

```
Files to Submit

1. palindrome.py
2. sin.py
3. distance.py
4. reverse.py
5. transpose.py

Before you submit your files, make sure:

- You do not use concepts from sections beyond “Libraries and Applications”.
- Your code is adequately commented, follows good programming principles, and meets any specific requirements such as corner cases and running times.