Exercise 3 (Collection Data Types and Standard Input)

Problem 1. (*Birthday Problem*) Suppose that people enter an empty room until a pair of people share a birthday. On average, how many people will have to enter before there is a match? Write a program called `birthday.py` that accepts *trials* (int) as command-line argument, runs *trials* experiments to estimate this quantity — each experiment involves sampling individuals until a pair of them share a birthday, and writes the value to standard output.

```
≻_ ~/workspace/exercise3

$ python3 birthday.py 1000
24
$ python3 birthday.py 1000
25
```

Problem 2. (*Pascal's Triangle*) Pascal's triangle $\mathcal{P}_n$ is a triangular array with $n+1$ rows, each listing the coefficients of the binomial expansion $(x+y)^i$, where $0 \le i \le n$. For example, $\mathcal{P}_4$ is the triangular array:

$$\begin{array}{lllll} 1 & & & & \\ 1 & 1 & & & \\ 1 & 2 & 1 & & \\ 1 & 3 & 3 & 1 & \\ 1 & 4 & 6 & 4 & 1 \end{array}$$

The term $\mathcal{P}_n(i,j)$ is calculated as $\mathcal{P}_n(i-1,j-1) + \mathcal{P}_n(i-1,j)$, where $0 \le i \le n$ and $1 \le j < i$, with $\mathcal{P}_n(i,0) = \mathcal{P}_n(i,i) = 1$ for all i. Write a program called `pascal.py` that accepts n (int) as command-line argument, and writes $\mathcal{P}_n$ to standard output.

```
≻_ ~/workspace/exercise3

$ python3 pascal.py 3
1
1 1
1 2 1
1 3 3 1
$ python3 pascal.py 10
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
```

Problem 3. (*Euclidean Distance*) Write a program called `distance.py` that accepts n (int) as command-line argument, two n-dimensional lists x and y of floats from standard input, and writes to standard output the Euclidean distance between two vectors represented by x and y. The Euclidean distance is calculated as the square root of the sums of the squares of the differences between the corresponding entries.

```
≻_ ~/workspace/exercise3

$ python3 distance.py 2
1 0 0 1 <enter>
1.4142135623730951
$ python3 distance.py 5
-9 1 10 -1 1 -5 9 6 7 4 <enter>
13.0
```

Problem 4. (*Reverse*) Write a program called `reverse.py` that accepts strings from standard input, and writes them in reverse order to standard output.

```
>_ ~/workspace/exercise3

$ python3 reverse.py
b o l t o n
<ctrl-d>
n o t l o b
$ python3 reverse.py
madam
<ctrl-d>
madam
```

Problem 5. (*Transpose*) Write a program called `transpose.py` that accepts m (int) and n (int) as command-line arguments, $m \times n$ floats from standard input representing the elements of an $m \times n$ matrix a, and writes to standard output the transpose of a.

```
>_ ~/workspace/exercise3

$ python3 transpose.py 2 2
1 2 3 4 <enter>
1.0 3.0
2.0 4.0
$ python3 transpose.py 2 3
1 2 3 4 5 6 <enter>
1.0 4.0
2.0 5.0
3.0 6.0
```

Files to Submit

1. `birthday.py`
2. `pascal.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 “Input and Output”.
- Your code is adequately commented, follows good programming principles, and meets any specific requirements such as corner cases and running times.