Problem 1. (*Quadratic Equation*) Write a program called `quadratic.py` (a variant of the program we discussed in class) that accepts a (float), b (float), and c (float) as command-line arguments, and writes to standard output the roots of the quadratic equation $ax^2 + bx + c = 0$. Your program should report the message "Value of a must not be 0" if $a = 0$, and the message "Value of discriminant must not be negative" if $b^2 - 4ac < 0$.

```
>_ ~/workspace/exercise2
$ python3 quadratic.py 1 -5 6
3.0 2.0
$ python3 quadratic.py 1 -1 -1
1.618033988749895 -0.6180339887498949
```

Problem 2. (*Six-sided Die*) Write a program called `die.py` that simulates the roll of a six-sided die, and writes to standard output the pattern on the top face.

```
>_ ~/workspace/exercise2
$ python3 die.py
*   *
  *
*   *
$ python3 die.py
*

    *
```

Problem 3. (*Primality Test*) Write a program called `primality_test.py` that accepts n (int) as command-line argument, and writes to standard output if n is a prime number or not.

```
>_ ~/workspace/exercise2
$ python3 primality_test.py 31
True
$ python3 primality_test.py 42
False
```

Problem 4. Write a program called `factorial.py` that accepts n (int) as command-line argument, and writes to standard output the value of $n!$, which is defined as $n! = 1 \times 2 \times \ldots (n-1) \times n$. Note that $0! = 1$.

```
>_ ~/workspace/exercise2
$ python3 factorial.py 0
1
$ python3 factorial.py 5
120
```

Problem 5. (*Counting Primes*) Write a program called `prime_counter.py` that accepts n (int) as command-line argument, and writes to standard output the number of primes less than or equal to n.

```
>_ ~/workspace/exercise2
$ python3 prime_counter.py 10
4
$ python3 prime_counter.py 100
25
```

Files to Submit

1. `quadratic.py`
2. `die.py`

3. `primality_test.py`
4. `factorial.py`
5. `prime_counter.py`

Before you submit your files, make sure:

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