Selected Exercises in Lua Programming

1. Create a set of geometric objects in Lua as tables. For example:

```
square1 = {side = 5}
square2 = {side = 7}
rectangle1 = {width = 4, height = 6}
rectangle2 = {width = 8, height = 2}
circle1 = {radius = 3}
triangle1 = {side1 = 5, side2 = 4, side3 = 3}
```

   Build one function to calculate the perimeter of a geometric figure and another to compute its area, such as in the source code below. Note that the function needs to identify the object from its properties.

```
print(area(square1))
print(perimeter(circle1))
```

2. Implement a function that takes an integer value as argument and invert the order of its numerals.

```
print(invert(531)) -- 135
```

3. Implement a function that gets a list of integer numbers and a number n and prints all the values smaller than n.

```
print(list({1,7,2,3,0,6}, 4)) -- 7 and 6
```

4. Given a list of integers, write a Lua function that extracts all odd numbers from it.

```
print(odds({1,7,2,3,0,6})) -- 1, 7, 3
```

5. Write a Lua function that calculates the factorial of a number n (the product of all numbers equals or smaller than n.

```
print(factorial(4)) -- 24
```

6. Write a Lua function that calculates all integers that are dividers of n.

```
print(dividers(4)) -- 1, 2, and 4
```

7. Write a Lua function that calculates the 50 first Fibonnaci numbers, defined as F(n) = F(n-1) + F(n-2) with F(0) = 0 and F(1) = 1.

```
print(fibonacci(7)) -- 8
```

8. Write a Lua function that calculates the 50 first prime numbers.

```
primesUntil50() -- 1, 2, 3, 5, ..., 47
```

9. Write a Lua function that calculates the 15 first Mersenne primes. A Mersenne prime is a number p such that 2^p - 1 is prime.

```
mersenne15()
```