Problem 1

Write a function `test_if_prime` that has some integer parameter, `n`, and returns a boolean corresponding to whether the parameter was prime, *e.g.,*

```
bool isprime;
int n;
isprime = test_if_prime(n);
```

Take your prime number testing function `test_if_prime`, and use it to write a program that prints multiple primes. Read an integer `how_many`, indicating how many *successive* prime numbers should be printed. Then print that many successive primes (each on a separate line). *Hint:* Keep a variable `number_of_primes_found` that is increased whenever a new prime is found.

Problem 2

Write a function with inputs `x`, `y`, and θ. Have this function alter the values of `x` and `y` corresponding to a rotation of the point (`x, y`) about an angle θ:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{1}$$

Your program should use the values `x = 1.0` and `y = 0.0` and accept any multiple of `pi` for θ (*e.g.,* a user input of "`2.0`" will correspond to $\theta = 2\pi$). The definition of `pi` will be useful:

```
#include <cmath>;
const float pi = 2*acos(0.0);
```

Output the new coordinate set as, *e.g.,*

```
x = 0.00, y = 1.00
```

if the user input was `0.5`.