This lab includes (i) binary algorithm for modular exponentiation and (ii) Eratosthenes algorithm for primality test. The API functions are defined as follows.

```
int modExp(int a, int e, int m)
// a: base number
// e: exponent
// m: modular number
// return: a^e mod m in normal cases (a, e, m > 0) and -1 otherwise

bool Eratosthenes(int n)
// n: target integer for primality test
// return: true if n is a prime number and false otherwise
```

Instructions

- Assume that your working directory (pwd) is `lab/`. Open and edit `src/lab.cc` to implement target functions.

```
$ vim src/lab.cc
```

- Compile your code using gtest framework.

```
$ make
```

- Run gtest.

```
$ ./gtest
```

- Revise your code based on promote information.

Promote Information

Three tests are evaluated for `modExp()`:

- *inputNegative:* the input a, e or m is not positive.
- *handbookCases:* failed in the examples illustrated in class.
- *otherCases:* failed in additional examples.

Three tests are evaluated for `Eratosthenes()`:

- *inputNegative:* the input n is negative.
- *trivalCases:* failed in trivial examples (e.g., n = 0, 1, 2, 3).
- *positiveCases:* failed in additional examples that n is positive.