Tristan Erney
November 21st, 2021
Intro to Cryptology
Hands on Exercise 11

#1

1	7	9	11	17	19	23	29
31	37	41	43	47		53	59
61	67	71	73		79	83	89
	97	101	103	107	109	113	
	127	131		137	139		149

```
#include <iostream>
#include <stdlib.h>
#include <stdint.h>
#include <cmath>
#include <vector>

int** prime_sieve(int n) {
  int J = floor(n / 30) + 1;
  int** primes = (int**)malloc(sizeof(int*) * J);
  for (int i = 0; i < J; i += 1) {
    primes[i] = (int*)malloc(sizeof(int) * 8);
    for (int j = 0; j < 8; j += 1) primes[i][j] = 0;
  }

  int K[] = { 1, 7, 11, 13, 17, 19, 23, 29 };
  for (int j = 0; j < floor(n / 30) + 1; j += 1) {
    for (int k = 0; k < 8; k += 1) {
      primes[j][k] = 30 * j + K[k];
    }
  }

  for (int j = 1; j < floor(n / 30) + 1; j += 1) {
    for (int k = 0; k < 8; k += 1) {
      int i = primes[j][k];
      if (i > 2 && i % 2 == 0) primes[j][k] = 0;
      if (i > 3 && i % 3 == 0) primes[j][k] = 0;
      if (i > 5 && i % 5 == 0) primes[j][k] = 0;
      if (i > 7 && i % 7 == 0) primes[j][k] = 0;
      if (i > 11 && i % 11 == 0) primes[j][k] = 0;
    }
  }

  return primes;
}

int main() {
```

```
  int** primes = prime_sieve(150);

  for (int i = 0; i < 5; i += 1) {
    for (int j = 0; j < 8; j += 1) {
      std::cout << primes[i][j] << ", ";
    }
    std::cout << "\n";
  }
  std::cout << "\n";

  free(primes);
  return 0;
}
```

#2

a.

$n = pq = 3(11) = 33$
$\phi(n) = (p - 1)(q - 1) = 2(10) = 20$

$C = m^e \pmod{n} = 5^7 \pmod{33} = 14$

$d * e = 1$ (mod phi)
$d * 3 = 1$ (mod phi)
$d = 3$

$m = C^d \pmod{n} = 14^3 \pmod{33} = 5$

b.

$n = pq = 5(11) = 55$
$\phi(n) = (p - 1)(q - 1) = 4(10) = 40$

$C = m^e \pmod{n} = 9^3 \pmod{55} = 14$

$d * e = 1$ (mod phi)
$d * 3 = 1 \pmod{40}$
$d = 27$

$m = C^d \pmod{n} = 14^{27} \pmod{55} = 9$

c.

$n = pq = 7(11) = 77$
$\phi(n) = (p - 1)(q - 1) = 6(10) = 60$

$C = m^e \pmod{n} = 8^{17} \pmod{77} = 57$

$d * e = 1$ (mod phi)
$d * 17 = 1$ (mod phi)

$d = 53$

$m = C^d \pmod n = 57^{53} \pmod{77} = 8$