```
class Primes {
 public:
  int getPrimeCount() const { return prime count; }
  int getPrime(int i) const { return primes[i]; }
  void addPrime(int i) { primes[prime count++] = i; }
  bool isDivisibe(int i, int by) { return (i % by) == 0; }
  bool isPrimeDivisible(int candidate) {
    for (int i = 1; i < prime count; ++i) {
      if (isDivisibe(candidate, primes[i])) return true;
    return false;
 private:
  volatile int prime count;
  volatile int primes[25000];
};int main() {
  Primes p;
  int c = 1;
  while (p.getPrimeCount() < 25000) {</pre>
    if (!p.isPrimeDivisible(c)) {
      p.addPrime(c);
    C++;
  printf("%d\n", p.getPrime(p.getPrimeCount()-1));
```