```
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) {
      if (!p.isPrimeDivisible(c)) {
        p.addPrime(c);
      }
      c++;
    }
    printf("%d\n", p.getPrime(p.getPrimeCount()-1));
  }
```