Cannot understand this prime generator algorithm in my textbook

I am studying Elements of Programming Interviews, and I am stuck on a problem. It is about writing a c++ function for finding all prime numbers from 1 to n, for a given n.

```
vector<int> generate_primes_from_1_to_n(const int &n) {
    int size = floor(0.5 * (n - 3)) + 1;
    // is_prime[i] represents (2i+3) is prime or not

    vector<int> primes; // stores the primes from 1 to n

    primes.push_back(2);
    vector<bool> is_prime(size, true);

    for(long i = 0; i < size; ++i) {
       if(is_prime[i]) {
           int p = (i << 1) + 3;
           primes.push_back(p);
           // sieving from p^2, whose index is 2i^2 + 6i + 3
           for (long j = ((i * i) << 1) + 6 * i + 3; j < size; j += p) {
               is_prime[j] = false;
           }
       }
    }
}
```

Particularly, I cannot understand the commented 'sieving from p^2, whose index is 2i^2 + 6i + 3' part. For the other parts, I can kind of grasp a rough idea of how they work, but I don't know where this '2i^2 + 6i + 3' comes from, what it does, and how that and its related pieces of codes work.

Can anyone explain this code better? Thank you.

+

I am getting this output(+'cout's to understand it better)

```
./a.out 100
size is: 49
for i = 0 is_prime[i] is 1
pushing back p of size 3
((i * i) << 1) + 6 * i + 3 for i of 0 is 3
((i * i) << 1) + 6 * i + 3 for i of 0 is 6
((i * i) << 1) + 6 * i + 3 for i of 0 is 9
((i * i) << 1) + 6 * i + 3 for i of 0 is 12
((i * i) << 1) + 6 * i + 3 for i of 0 is 15
((i * i) << 1) + 6 * i + 3 for i of 0 is 18
((i * i) << 1) + 6 * i + 3 for i of 0 is 21
((i * i) << 1) + 6 * i + 3 for i of 0 is 24
((i * i) << 1) + 6 * i + 3 for i of 0 is 27
((i * i) << 1) + 6 * i + 3 for i of 0 is 30
((i * i) << 1) + 6 * i + 3 for i of 0 is 33
((i * i) << 1) + 6 * i + 3 for i of 0 is 36
((i * i) << 1) + 6 * i + 3 for i of 0 is 39
((i * i) << 1) + 6 * i + 3 for i of 0 is 42
((i * i) << 1) + 6 * i + 3 for i of 0 is 45
((i * i) << 1) + 6 * i + 3 for i of 0 is 48
```