|  |  |
| --- | --- |
| **Seive in C++** | |
| #include <iostream>  #include <cmath>  #include <vector>  using namespace std;  class SeiveofErastostenins {  public:  static void main() {  vector<bool> myseive = seive(20);  for (int i = 0; i < myseive.size(); i++) {  cout << i << " " << (myseive[i] ? "true" : "false") << endl;  }  }  static vector<bool> seive(int n) {  vector<bool> arr(n + 1, true);  arr[0] = false;  arr[1] = false;  for (int i = 2; i <= sqrt(n); i++) {  if (arr[i]) {  for (int j = i \* i; j <= n; j += i) {  arr[j] = false;  }  }  }  return arr;  }  };  int main() {  SeiveofErastostenins::main();  return 0;  } | **Sieve of Eratosthenes Dry Run for n = 20**  **🔢 Step 1: Initialize Boolean Vector**  vector<bool> arr(n + 1, true); // arr[0..20] all set to true  arr[0] = false;  arr[1] = false;  **📋 Initial Table:**   | **i** | **isPrime** | | --- | --- | | 0 | false | | 1 | false | | 2 | true | | 3 | true | | 4 | true | | 5 | true | | 6 | true | | 7 | true | | 8 | true | | 9 | true | | 10 | true | | 11 | true | | 12 | true | | 13 | true | | 14 | true | | 15 | true | | 16 | true | | 17 | true | | 18 | true | | 19 | true | | 20 | true |   **🔁 Step 2: Outer loop — for (int i = 2; i <= sqrt(n); i++)**   * sqrt(20) is ~4.47 → so i goes from 2 to 4   **➤ i = 2:**  arr[2] == true → mark all multiples of 2 from 4 onward as false  **Inner loop (j = i\*i; j <= n; j += i) → j = 4, 6, 8, 10, 12, 14, 16, 18, 20**  **❌ Marked False:**  4, 6, 8, 10, 12, 14, 16, 18, 20  **➤ i = 3:**  arr[3] == true → mark all multiples of 3 from 9 onward as false  **j = 9, 12, 15, 18**  **❌ Marked False:**  9, 15  (12 and 18 already marked by i = 2)  **➤ i = 4:**  arr[4] == false → skip  **✅ Final Table After Sieve:**   | **i** | **isPrime** | | --- | --- | | 0 | false | | 1 | false | | 2 | true | | 3 | true | | 4 | false | | 5 | true | | 6 | false | | 7 | true | | 8 | false | | 9 | false | | 10 | false | | 11 | true | | 12 | false | | 13 | true | | 14 | false | | 15 | false | | 16 | false | | 17 | true | | 18 | false | | 19 | true | | 20 | false |   **🧾 Output Printed by the Code:**  0 false  1 false  2 true  3 true  4 false  5 true  6 false  7 true  8 false  9 false  10 false  11 true  12 false  13 true  14 false  15 false  16 false  17 true  18 false  19 true  20 false  **✅ Prime Numbers ≤ 20:**  **2, 3, 5, 7, 11, 13, 17, 19** |
| 0 false  1 false  2 true  3 true  4 false  5 true  6 false  7 true  8 false  9 false  10 false  11 true  12 false  13 true  14 false  15 false  16 false  17 true  18 false  19 true  20 false | |