

## primeNumbersCollector.cpp

```

1  #include <iostream>
2  #include <vector> //so can add elements to arrays at runtime
3  #include <cmath>
4
5  using namespace std;
6  int desiredDigits = 3;
7
8  vector<int> primeFinder(int digitCount){
9      int maxNum = pow(10,digitCount) - 1;
10     //for example, if want 3 digits, then get 10^3 -1 = 1000 -1 = 999, the largest 3 digit
    numner
11     vector<int> primes = {2,3};
12
13     //cout << "maxNum: " << maxNum << " \n";
14     int ii; //need to declare this outside the loop so that stuff outside the loop can access
    it
15
16     //sizeof(primes)/sizeof(primes[0]) is the length of the array, primes with plain c++
    arrays
17     //primes.size() is how it is done with vectors in c++
18     //will start collecting at the end of the primes list + 1, so in this case, at 3+1=4
19     for(int i = primes[primes.size() - 1] + 1; i<=maxNum ; i++){
20         //cout << "i = " << i << " \n";
21         for(ii = 0;ii<primes.size();ii++){
22             if(i % primes[ii] == 0){
23                 break;
24                 //modulo to check for remainders of dividing i by primes(ii)
25                 //if remainder is 0, then that number is divisible, so break
26                 //loop, and skip this number
27             }
28         }
29         //cout << "checked, ii = " << ii << " , length of primes: " << primes.size() << " \n";
30         //if it reached the end of the list without breaking loop (for ii), then
31         //that number was not found to be divisible by anything, so its prime
32         //this works because you keep adding more and more numbers to the list,
33         //so bigger numbers have more factors to check against
34         if(ii == primes.size()){
35             primes.push_back(i);
36         }
37     }
38     return primes;
39 }
40
41 int main(){
42     int sumOfPrimes=0;
43     vector<int> foundPrimes = primeFinder(desiredDigits);
44     cout << "Primes Up To " << desiredDigits << " Digits: \n";
45     for (const int &i : foundPrimes) {
46         cout << i << " , ";
47         sumOfPrimes += i;
48     }
49     cout << "\n";
50     cout << "Sum Of Primes Up To " << desiredDigits << " Digits: "<< sumOfPrimes << "\n";

```

```
51 |     return 0;  
52 | }  
53 |  
54 |
```