

```

1 // -----以下ライブラリ-----//
2 //O(N)
3
4 #include <iostream>
5 #include <utility>
6 #include <map>
7 #include <string>
8 #include <vector>
9 #include <algorithm>
10 #include <functional>
11 #include <array>
12 #include <math.h>
13 #include <numeric>
14 #include <sstream>
15 typedef long long ll;
16 using namespace std;
17
18 vector<int> primes;
19 map<int, int> factors;
20
21 void erathosthenes(int n) {
22     bool primeFlags[n+1]; //boolでふるいを用意する。素数であればtrueであるとする
23     for (int i = 0; i <= n; i++) primeFlags[i] = true; //ふるいをリセット
24     primeFlags[0] = primeFlags[1] = false; //0と1は素数ではないのでふるい落とす(=falseにする)
25     vector<int> sqrtprimes; //ルートnまでの素数を入れておくための配列
26     for (int i = 2; i*i <= n; i++) {
27         bool mod0 = false;
28         for (int j = 2; j*j <= i; j++) {
29             if (i%j == 0) mod0 = true;
30         }
31         if (!mod0) sqrtprimes.push_back(i);
32     }
33     for (int i = 0; i < sqrtprimes.size(); i++) {
34         for (int j = 2; j <= n; j++) {
35             if (j%sqrtprimes[i] == 0) primeFlags[j] = false;
36         }
37     }
38     for (int i = 0; i < sqrtprimes.size(); i++) primeFlags[sqrtprimes[i]] = true;
39     for (int i = 1; i <= n; i++) if (primeFlags[i]) primes.push_back(i);
40     return;
41 }
42
43 void factoring(int n) {
44     for (int i = 0; i < primes.size(); ++i) {
45         while(n%primes[i] == 0){
46             factors[primes[i]]++;
47             n /= primes[i];
48         }
49     }
50 }
51
52 void factorial(int n) {
53     erathosthenes(n);
54     for (int i = 2; i <= n; ++i) {
55         factoring(i);
56     }
57 }
58
59 //-----以上ライブラリ-----//-----以下debug用main関数-----//
60

```

```
61
62 int main(int argc, char const *argv[]) {
63     int n;
64     cin >> n;
65     factorial(n);
66     cout << "Primes =" << " ";
67     for (auto x : primes) cout << x << " ";
68     cout << "" << std::endl;
69     string factorString = "factoring = ";
70     for (auto x : factors) factorString += to_string(x.first) + "^" + to_string(x.second) +
" * ";
71     for (int i = 0; i < 3; i++) factorString.pop_back();
72     cout << factorString << std::endl;
73     return 0;
74 }
75
```