

```
1 // -----以下ライブラリ-----//
2
3 #include <iostream>
4 #include <utility>
5 #include <map>
6 #include <string>
7 #include <vector>
8 #include <algorithm>
9 #include <functional>
10 #include <array>
11 #include <math.h>
12 #include <numeric>
13 #include <sstream>
14 typedef long long ll;
15 using namespace std;
16
17 vector<int> primes;
18 map<int, int> factors;
19
20 void erathosthenes(ll n) {
21     vector<bool> primeFlags(n+10, true);
22     primeFlags[0] = primeFlags[1] = false;
23     vector<int> sqrtprimes;
24     for (int i = 2; i*i <= n; i++) {
25         bool mod0 = false;
26         for (int j = 2; j*j <= i; j++) {
27             if (i%j == 0) mod0 = true;
28         }
29         if (!mod0) sqrtprimes.push_back(i);
30     }
31     for (auto &&e: sqrtprimes) {
32         for (int i = 2; i <= n; i++) {
33             if (i%e == 0) primeFlags[i] = false;
34         }
35     }
36     for (auto &&e: sqrtprimes) primeFlags[e] = true;
37     for (int i = 1; i <= n; i++) if (primeFlags[i]) primes.push_back(i);
38     return;
39 }
40
41 void factoring(ll n) {
42     ll m = sqrt(n);
43     erathosthenes(m);
44     for (auto &&e: primes) {
45         while(n%e == 0){
46             factors[e]++;
47             n /= e;
48         }
49     }
50     factors[n]++;
51     factors.erase(1);
52 }
53
54
55
56 //-----以下debug用main関数-----//
57
58
59 int main(int argc, char const *argv[]) {
60     int n;
```

```
61     cin >> n;
62     factoring(n);
63     cout << "Primes =" << " ";
64     for (auto x : primes) cout << x << " ";
65     cout << "" << std::endl;
66     string factorString = "factoring = ";
67     for (auto &&x : factors) factorString += to_string(x.first) + "^" + to_string(x.second)
+ " * ";
68     for (int i = 0; i < 3; i++) factorString.pop_back();
69     cout << factorString << std::endl;
70     return 0;
71 }
```