**★★★★☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：11099:Next Same-Factored**

**解題者：郭昱**

**解題日期：2021年6月9日**

**題意：**

輸入一整數 n (0<n<1000000)，S為n的所有質因數集合。求比n大，且由相同質因數集合S所組成的下一個整數，若所求大於2000000或是不存在，則輸出”Not Exist!”。

**題意範例：**

1. 🡺 Not Exist

6 🡺 2 \* 3 🡺 12

42 🡺 2 \* 3 \* 7 🡺 84
3571🡺 3571 🡺 Not Exist

100 🡺 (2^2) \* (5^2) 🡺 160

6897🡺 3 \* (11^2) \* 19 🡺 11913

**解法：**

(1)建質數表 (1~1000000)

 (2)找出n的質因數

 (3)用priority queue實作BFS找n的next same-factored

**解法範例：**

**討論：**

(1)重複的情況 🡺 紀錄上一個base，重複便不計算

 (2) Not Exist 🡺 超過2000000不push，queue空了= Not Exist

**程式：**

#include<iostream>

#include<cmath>

#include<queue>

using namespace std;

int prime(int n)

{

 if(n==2)

 return 1;

 if(n<=1)

 return 0;

 for(int i=2; i<=(int)sqrt(n); i++)

 {

 if(n%i==0)

 return 0;

 }

 return 1;

}

int prime\_table[78498];

void make\_table()

{

 int sum=0, flag;

 prime\_table[sum++] = 2;

 for(int i=3; i<=999983; i+=2)

 {

 flag = 1;

 for(int j=3; j<=(int)sqrt(i); j++)

 {

 if(i%j==0)

 {

 flag=0;

 break;

 }

 }

 if(flag)

 prime\_table[sum++] = i;

 }

 //cout << sum << endl;

}

int factor\_table[100][2];

void factorization(int n)

{

 int count=0;

 factor\_table[0][0] = factor\_table[1][0] = 0;

 for(int i=n, k=0; i>1; k++)

 {

 if(i%prime\_table[k] == 0)

 {

 //return i\*prime\_table[k];

 if(factor\_table[count][0]!=prime\_table[k])

 {

 factor\_table[++count][0]=prime\_table[k];

 factor\_table[count+1][0] = 0;

 factor\_table[count][1] = 0;

 }

 factor\_table[count][1]++;

 //cout << prime\_table[k] << " ";

 i /= prime\_table[k--];

 }

 }

 //cout << endl;

}

void BFS(int n)

{

 priority\_queue<int, vector<int>, greater<int> > pool;

 int base=1;

 for(int i=1; factor\_table[i][0]!=0; i++)

 base \*= factor\_table[i][0];

 pool.push(base);

 int flag = 1;

 while(base <= n && !pool.empty())

 {

 base = pool.top();

 pool.pop();

 for(int i=1; factor\_table[i][0]!=0; i++)

 {

 if((float)base\*factor\_table[i][0] <= 2000000)

 pool.push(base\*factor\_table[i][0]);

 else

 {

 break;

 }

 }

 }

 if(base<=2000000 && base>n)

 cout << base << endl;

 else

 cout << "Not Exist!\n";

}

void next\_factor(int n)

{

 if(n==1)

 {

 cout << "Not Exist!\n";

 return;

 }

 int sum, i;

 for(i=1, sum=0; factor\_table[i][0]!=0; i++)

 if(factor\_table[i][1]==1)

 sum++;

 if( i==sum+1 )

 if( (double)n \* factor\_table[1][0] <= 2000000)

 cout << n \* factor\_table[1][0] << endl;

 else

 cout << "Not Exist!\n";

 else

 BFS(n);

}

int main()

{

 make\_table();

 int n;

 while(cin >> n)

 {

 factorization(n);

 next\_factor(n);

 }

}