/\*

N. 【基础】求无暇素数

题目描述

一个两位整数A本身是素数，若将其个位数字与十位数字交换，得到一个新的两位数B，而B也是素数，我们则称A为无暇素数。 例如：31是素数，个位数字与十位数字交换后得到13，也是素数。所以31是无暇素数。 问题： 给出一个数字字符串（即字串中的字符全部由数字组成），求出其所有组成的无暇素数。

输入

一个数字字符串（长度<＝20），以"%"结束。

输出

全部的无暇素数，之间用一个逗号隔开。

样例

输入复制

321314%

输出复制

13,31

\*/

#include <bits/stdc++.h>

using namespace std;

bool sushu(int x)

{

int c = 0;

for (int i = 0; i <= sqrt(x); i++)

{

if (x % i == 0)**//当i=0时，0不能做除数，代码不报错，但执行不成功。i可以从2开始**

{

c++;

return false;

break;**//程序遇到return时会返回后面的值，函数结束，不会执行break。可以删掉break**

}

}

if (c == 0 && x != 1)

{

return true;

}**//return都写在if里面了，程序找不到return，警告提示：warning: control reaches end of non-void function [-Wreturn-type]，可以给默认返回return值。**

}

bool wuxia(int x)

{

cout << x;

int a = x, b = x / 10 + x % 10 \* 10;

if (sushu(a) && sushu(b))

{

return 1;

}

else

{

return 0;

}

}

int main()

{

string n;

getline(cin, n);

string c = "";

int q, a = 0;

for (int i = 1; i < n.size(); i++)**//字符串结尾是%，i循环范围需要减1**

{

c += n[i - 1];

c += n[i];

q = stoi(c);

if (wuxia(q))

{

if (a == 0) **//a判断完了要++**

{

cout << q;

}

else

{

cout << ',' << q;

}

}

c = "";

}

// cout <<c;

return 0;

}

修改后的代码如下：

#include <bits/stdc++.h>

using namespace std;

bool sushu(int x) {

  for (int i = 2; i <= sqrt(x); i++) {

    if (x % i == 0 || x == 1) return 0;

  }

  return 1;

}

bool wuxia(int x) {

  int a = x, b = x / 10 + x % 10 \* 10;

  if (sushu(a) && sushu(b))

    return 1;

  else

    return 0;

}

int main() {

  string n;

  getline(cin, n);

  string c;

  int q, a = 0;

  for (int i = 1; i < n.size() - 1; i++) {

    c += n[i - 1];

    c += n[i];

    q = stoi(c);

    if (wuxia(q)) {

      if (a == 0) {

        cout << q;

      } else {

        cout << ',' << q;

      }

      a++;

    }

    c = "";

  }

  return 0;

}