C++ 第七章作业一

题目描述

编写程序，实现单向链表结构。为每个输入的素数创建新节点，并按照从小到大的顺序插入到链表中的适当位置。当输入非素数时，停止创建新节点并将链表内的内容从首至尾遍历输出。请注意，输出内容严格单增，即输出的元素不能重复。

输入

任意n个整数，彼此间用空格分隔，整数范围[-10000, 10000]，n < 100。

输出

m个素数，并且保持严格单增（m < n）

样例输入

23 5 37 31 17 5 23 28

样例输出

5 17 23 31 37

提示

需要使用类创建节点对象类型。

#include<iostream>

#include<cmath>

using namespace std;

class node

{

public:

int num;

node\* next;

node(int n)

{

num=n;

next = nullptr;

}

~node();

};

class list

{

public:

node\* head;

node\* tail;

list()

{

head = nullptr;

tail = nullptr;

}

bool same(int n)//遍历链表查找是否有重复的数据

{

node\* cur = head;

if (cur == nullptr)

{

return true;

}

else

{

cur = cur->next;

while (cur!= nullptr)

{

if (cur->num == n)

{

return false;//有重复

}

cur = cur->next;

}

return true;//无重复

}

}

void insert(int n)//若无重复且为素数，则插入数据

{

node\* temp = new node(n);

if (head == nullptr)

{

head = tail = temp;

}

else

{

if (n < head->num)

{

temp->next = head;

head = temp;

return;

}

if (n > tail->num)

{

tail->next = temp;

tail = temp;

return;

}

node\* cur = head;

while (cur->next != nullptr)

{

if (cur->num<n && cur->next->num>n)

{

temp->next = cur->next;

cur->next = temp;

return;

}

cur = cur->next;

}

}

}

void print()//输出list中的数据

{

node\* cur = head;

if (cur == nullptr)

{

return;

}

else

{

cout << cur->num;

cur = cur->next;

while (cur != nullptr)

{

cout << " " << cur->num;

cur = cur->next;

}

}

}

};

bool sushu(int n)//判断是否是素数

{

if (n <= 1)

{

return false;//不是素数

}

else

{

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

{

if (n % i == 0)

{

return false;//不是素数

}

}

return true;//是素数

}

}

int main()

{

list l;

int n;

while (cin >> n)

{

if (sushu(n) && l.same(n))

{

l.insert(n);

}

else

{

break;

}

}

l.print();

return 0;

}

C++ 第七章作业二

题目描述

建立一个对象数组，内放4个学生的数据（学号 int、成绩 int），设立一个函数max()，用

指向对象的指针作函数参数，在max函数中找出4个学生中成绩最高者，并输出其学号和成绩。如有相同最高成绩，输出所有符合的学号和成绩。

输入

输入8个整数：学号1 成绩1 学号2 成绩2... 用空格分隔。

输出

最高成绩和对应的学号

样例输入

3 100 2 90 4 78 5 100

样例输出

3 100

5 100

#include<iostream>

using namespace std;

class student

{

public:

int num;

int score;

void setnum(int n)

{

num = n;

}

void setscore(int n)

{

score = n;

}

int max(student\* cur)

{

int max = cur[0].score;

int k = 0;

for (int i = 0; i < 4; i++)

{

if (cur[i].score > max)

{

max = cur[i].score;

k = i;

}

}

return cur[k].score;

}

};

int main()

{

student stu[4];

for (int i = 0; i < 4; i++)

{

int a, b;

cin >> a >> b;

stu[i].setnum(a), stu[i].setscore(b);

}

int max = stu[0].max(stu);

for (int i = 0; i < 4; i++)

{

if (stu[i].score == max)

{

cout << stu[i].num << " " << stu[i].score << endl;

}

}

return 0;

}