**NetGames**

时间限制(普通/Java):1000MS/3000MS          运行内存限制:65536KByte

总提交:88            测试通过:12

**描述**

给定表示单个游戏的类 Game 和 表示一组游戏的类 NetGames 的头文件部分，要求实现这两个类并写一个main函数按照要求测试这两个类。 测试要求：首先输入一个整数n，表示接下来会输入n个游戏名称，根据输入的这n个游戏名称生成n个Game对象，并把它们添加到一个NetGames对象ng中； 接下来，再输入一个游戏名称，去ng中查找这个游戏是否在ng中，如果在，输出"Yes"；否则，输出"No"

class Game { // a game

public:

 Game(string n = "");

 ~Game();

 string get\_name();

private:

 string name; // game name

};

class NetGames { // netgames

public:

 void add\_game(Game); // add a game to netgames

 bool search(Game); // search a game, if it is in the netgames, return true, else false

private:

 Game games[100];

 static int total\_number; // total number of games

};

**输入**

输入首先是一个整数n(1<=n<=100)，表示接下来会输入n个游戏名称(游戏名称中不包含空格)；最后再输入一个待查找的游戏名称（输入共n+1个游戏名称）

**输出**

输出 Yes 或 No

**样例输入**

5 WorldofWarcraft LeagueofLegends StarCraft Dota2 Vindictus StarCraft

**样例输出**

Yes

**提示**

按照要求答题，不按照要求过题不得分

**TV**

时间限制(普通/Java):1000MS/3000MS          运行内存限制:65536KByte

总提交:71            测试通过:20

**描述**

给定三个类的头文件部分，这三个类分别是表示电视剧的类 TV 以及 两个继承自TV的类 TeleStationTV（表示电视台制作的电视剧类）和 NetTV（表示网络平台制作的电视剧类）， 要求实现这三个类并用给定的main函数测试这三个类。 其中main函数中调用了一个work函数（全局函数），要求实现这个函数完成测试。以下是三个类的头文件和main函数。

class TV{

public:

 TV(string name="", string country="", int year=0);

 virtual void input();

 virtual void show();

 virtual ~TV();

protected:

 string name;  // 电视剧名称

 string country;  // 国别

 int year;         // 播出年份

};

class TeleStationTV : public TV {

public:

 TeleStationTV(string name="", string country="", int year=0, string station="");

 ~TeleStationTV();

 void input();

 void show();

private:

 string station;   // 电视台

};

class NetTV : public TV {

public:

 NetTV(string name="", string country="", int year=0, string net="");

 ~NetTV();

 void input();

 void show();

private:

 string net;  // 网络平台

};

int main() {

 char c;

 while (cin >> c) {

  switch (c) {

   case 'T':

    work(new TV);

    break;

   case 'S':

    work(new TeleStationTV);

    break;

   case 'N':

    work(new NetTV);

    break;

   default:

    break;

  }

 }

 return 0;

}

**输入**

输入包含多组数据。每一组数据首先是一个字母（T 或 S 或 N），如果是字母T，构造一个TV类对象，接下来是这个对象的信息的输入（分别是电视剧的名称name、国别country和播出年份year）； 如果是字母S，构造一个TeleStationTV对象，接下来是这个对象的信息的输入（分别是电视剧的名称、国别、播出年份以及制作的电视台station）； 如果是字母N，构造一个NetTV对象，接下来是这个对象的输入（分别是电视剧的名称、国别、播出年份以及制作的网路平台net）

**输出**

对应每组输入，输出对应的对象的信息，每组输出占一行，对象的成员之间以空格隔开

**样例输入**

T LangYaBang China 2015 S PersonOfInterest USA 2011 CBS N YuZui China 2016 iqiyi

**样例输出**

LangYaBang China 2015 PersonOfInterest USA 2011 CBS YuZui China 2016 iqiyi

**提示**

按照要求答题，不按照要求过题不得分；work函数的实现必须通过构造的对象调用input和show函数完成题目的要求

**Rational**

时间限制(普通/Java):1000MS/3000MS          运行内存限制:65536KByte

总提交:103            测试通过:23

**描述**

给定有理数类Rational(不考虑分母为0的情况)的头文件部分，要求实现这个类并用给定的main函数测试这个类。

class Rational {

private:

 int numerator; // 分子

 int denominator; // 分母

 void simplify(); // 化简

public:

 Rational();

 Rational(int n, int d);

 friend Rational operator+(const Rational&, const Rational&);

 friend Rational operator-(const Rational&, const Rational&);

 friend Rational operator\*(const Rational&, const Rational&);

 friend Rational operator/(const Rational&, const Rational&);

 friend ostream& operator<<(ostream&, const Rational&);

 friend istream& operator>>(istream&, Rational &);

 ~Rational();

};

int main() {

 Rational a,b;

 while(cin>>a>>b){

  cout << a + b << endl;

  cout << a - b << endl;

  cout << a \* b << endl;

  cout << a / b << endl;

 }

 return 0;

}

**输入**

输入包含多组数据，每组数据包含两行，每行两个整数，分别表示一个有理数的分子和分母

**输出**

对应每组输入，产生四行输出，分别表示输入的两个有理数的加、减、乘、除的结果 （如果有理数为0，直接输出0，否则输出化简了的：分子/分母, 如果有理数为负数，负号放在分子输出）

**样例输入**

1 2 1 2 3 4 5 6

**样例输出**

1/1 0 1/4 1/1 19/12 -1/12 5/8 9/10

**提示**

按照要求答题，不按照要求过题不得分

**题目来源**

NUAA

**Queue**

时间限制(普通/Java):1000MS/3000MS          运行内存限制:65536KByte

总提交:49            测试通过:15

**描述**

实现给定的Queue类模板并用给定的main函数测试这个类模板。

template<class T, int SIZE = 20>

class Queue{

private:

 T array[SIZE];        //数组，用于存放队列的元素

 int tail;              //队尾位置（数组下标）

public:

 Queue();                //构造函数，初始化队列

 void push(const T & );  //元素入队

 T pop();                //队首元素出队，后面的元素依次往前移

 void clear();           //将队列清空

 const T & head() const; //访问队首元素

 bool empty() const;     //测试队列是否为空

 bool full() const;      //测试队列是否满

 int size() const;       //返回当前队列中的元素个数

};

int main()

{

 Queue<int,10> intQueue;

 for(int i=0;i<10;i++)

  intQueue.push(i);

 if(intQueue.full()) cout<<"Now, intQueue is full."<<endl;

 cout<<intQueue.head()<<endl;

 for(int i=0;i<10;i++)

  cout<<intQueue.pop()<<" ";

 cout<<endl;

 if(intQueue.empty()) cout<<"Now, intQueue is empty."<<endl;

 Queue<string,5> stringQueue;

 stringQueue.push("One");

 stringQueue.push("Two");

 stringQueue.push("Three");

 stringQueue.push("Four");

 stringQueue.push("Five");

 cout<<"There are "<<stringQueue.size()<<" elements in stringQueue."<<endl;

 stringQueue.clear();

 if(stringQueue.empty()) cout<<"Now, there are no elements in stringQueue."<<endl;

 return 0;

}

**输入**

无

**输出**

Now, intQueue is full. 0 0 1 2 3 4 5 6 7 8 9  Now, intQueue is empty. There are 5 elements in stringQueue. Now, there are no elements in stringQueue.

**样例输入**

**样例输出**

Now, intQueue is full. 0 0 1 2 3 4 5 6 7 8 9  Now, intQueue is empty.

**题目来源**

NUAA

**StringSort**

时间限制(普通/Java):1000MS/3000MS          运行内存限制:65536KByte

总提交:99            测试通过:26

**描述**

对一组字符串先反转、之后按照降序输出。比如字符串"abc"，反转后就是"cba"。所有的字符串先反转，之后按照它们的降序（字典序）依次输出。

**输入**

第一行为一个整数N，表示字符串的数目。(0<N<50) 接下来是N行，每行一个字符串，每个字符串长度不超过100，所有字符串均不相同, 字符之间没有空格。

**输出**

排完序以后的字符串，每个字符串占一行

**样例输入**

5 happy nice good wonderful excellent

**样例输出**

yppah tnellecxe lufrednow ecin doog

**提示** 考虑使用STL的容器和算法

**File**

时间限制(普通/Java):1000MS/3000MS          运行内存限制:65536KByte

总提交:48            测试通过:0

**描述**

文件题目要求： 文件Mars.txt（这个文件根据下面的内容自己创建）中有若干单词（不超过500个），读取其中的单词，以10个一行输出到文件newMars.txt中，每一行中的单词以空格分隔。

文件Mars.txt的内容为：

The planet Mars has gone through an ice age and is now warming up, scientists say. The findings are based on detailed radar measurements of ice caps on Mars.

Mars is often called the red planet. But, a few hundred thousand years ago, the scientists said, the planet was not so red.

Their research suggests that Mars has been warming for the past 370,000 years.

Smith said ice may have built up to a thickness of several meters around Mars during its ice ages. This would have made the red planet look more white than red. Smith told VOA an ice age on Mars would be different than an ice age on Earth. He said, during Martian ice ages, the north and south poles would be warmer than they are now, and they would have less ice.

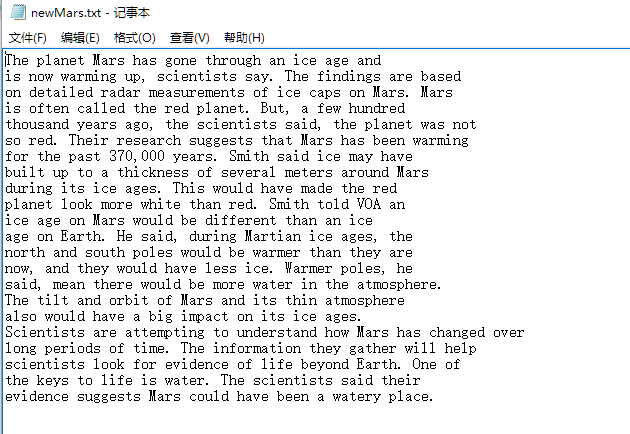
Warmer poles, he said, mean there would be more water in the atmosphere. The tilt and orbit of Mars and its thin atmosphere also would have a big impact on its ice ages.

Scientists are attempting to understand how Mars has changed over long periods of time.

The information they gather will help scientists look for evidence of life beyond Earth.

One of the keys to life is water. The scientists said their evidence suggests Mars could have been a watery place.

文件newMars.txt应如下所示:



**输入**

无

**输出**

无

**样例输入**

**样例输出**

**题目来源**

NUAA