/\*

习题11-3 使用I/O流以文本方式建立一个文件test1.txt，写入字符"已成功写入文件!"

用其他字处理程序打开，看看是否正确写入

\*/

#include<fstream>

#include<cstring>

using namespace std;

int main()

{

    ofstream file1("test1.txt");//使用code runner

    file1<<"已成功写入文件!";

    file1.close();

    return 0;

}

/\*

习题11-4 使用I/O流以文本方式打开11-3题建立的文件test1.txt

读出其内容并显示出来，看看是否正确

\*/

#include<iostream>

#include<fstream>

#include<cstring>

#include<cassert>

using namespace std;

int main()

{

    string a;

    ifstream file1("test1.txt");

    assert(file1);

    file1>>a;

    file1.close();

    cout<<a<<endl;

    return 0;

}

/\*

如出现乱码，点击右下角utf8，将其切换到GBK模式

在终端输入 chcp 查看编码 936代表GBK,65001代表utf8,然后输入chcp 65001即可切换

\*/

/\*

习题11-5 使用I/O流以文本方式打开11-3题建立的文件test1.txt

在文件后面添加字符"已成功添加字符!"

然后读出整个文件的内容显示出来，看看是否正确

\*/

#include<fstream>

#include<cstring>

using namespace std;

int main()

{

    ofstream file1("test1.txt", ios\_base::app);

    file1<<"已成功写入文件!";

    file1.close();

    return 0;

}

/\*

习题11-6 定义一个Dog类，包含体重和年龄两个成员变量及相应的成员函数

声明一个实例dog1，体重为5，年龄为10，使用I/O流把dog1的状态写入磁盘文件

再声明一个实例dog2，通过读文件把dog1的状态赋给dog2

分别使用文本方式和二进制方式操作文件，看看结果有何不同

再看看磁盘文件的ASCII码有何不同

\*/

#include<iostream>

#include<fstream>

#include<cstring>

#include<cassert>

using namespace std;

class Dog

{

private:

    int weight;

    int age;

public:

    Dog();

    ~Dog();

    Dog(int w, int a);

    Dog(const Dog& dog);

    int getWeight() const;

    int getAge() const;

};

Dog::Dog()

{

}

Dog::~Dog()

{

}

Dog::Dog(int w, int a)

:weight(w), age(a)

{

}

Dog::Dog(const Dog& dog)

:weight(dog.weight), age(dog.age)

{

}

int Dog::getWeight() const

{

    return weight;

}

int Dog::getAge() const

{

    return age;

}

int main()

{

    Dog dog1(5, 10);

    string filename;

    cout<<"请输入文件名: ";

    cin>>filename;

    ofstream file1(filename, ios\_base::binary);

    file1.write(reinterpret\_cast<char\*>(&dog1), sizeof(dog1));

    file1.close();

    ifstream file2(filename, ios\_base::binary);

    assert(file2);

    Dog dog2 = dog1;

    file2.read(reinterpret\_cast<char\*>(&dog2), sizeof(dog2));

    cout<<dog2.getWeight()<<", "<<dog2.getAge()<<endl;

    return 0;

}

//习题11-7 观察下面的程序，说明每条语句的作用，写出程序执行的结果

#include<iostream>

using namespace std;

int main()

{

    ios\_base::fmtflags original\_flags=cout.flags();

    cout<<812<<"|";

    cout.setf(ios\_base::left, ios\_base::adjustfield);

    cout.width(10);

    cout<<813<<815<<'\n';

    cout.unsetf(ios\_base::adjustfield);

    cout.precision(2);

    cout.setf(ios\_base::uppercase|ios\_base::scientific);

    cout<<831.0;

    cout.flags(original\_flags);

    return 0;

}

/\*

习题11-8 编写程序提示用户输入一个十进制整数，分别用十进制、八进制和十六进制形式输出

\*/

#include<iostream>

using namespace std;

int main()

{

    int n;

    cout<<"Please enter an integer: ";

    cin>>n;

    cout<<"Dec: "<<dec<<n<<endl;

    cout<<"Oct: "<<oct<<n<<endl;

    cout<<"Hex: "<<hex<<n<<endl;

    return 0;

}

/\*

习题11-9 编写程序实现如下功能：

打开指定的一个文本文件，在每一行前加行号后将其输出到另一个文本文件中

\*/

#include<iostream>

#include<fstream>

#include<cstring>

using namespace std;

int main()

{

    ifstream file1("test1.txt");

    ofstream file2("test2.txt");

    string line;

    int i=1;

    while (! getline(file1,line).eof())

    {

        file2<<i++<<" "<<line<<endl;

    }

    file2<<i++<<" "<<line<<endl;

    return 0;

}

/\*

习题11-10 使用宽输入流从一个有中文字符的文本文件中读入所有字符

统计每个字符出现的次数，将统计结果用宽输出流输出到另一个文本文件中

\*/

#include <iostream>

#include <fstream>

#include <cstring>

#include <map>

using namespace std;

int main()

{

    ifstream file1("test1.txt");

    ofstream file2("test2.txt");

    string line, temp;

    int i;

    map<string, int> counter;//UTF-8使用三个字节，所以将宽字符换成三字符

    map<string, int>::iterator iter;

    while(getline(file1, line))

    {

        for(i=0; i<line.size(); i+=3)

        {

            temp = "";

            temp += line[i];

            temp += line[i+1];

            temp += line[i+2];

            counter[temp]++;

        }

    }

    for(iter=counter.begin(); iter!=counter.end(); iter++)

    {

        file2<<iter->first<<", "<<iter->second<<endl;

    }

    file1.close();

    file2.close();

    return 0;

}