第十三周作业+答案

一．填空题

[1.1] 头文件 iostream 中定义了 4 个标准流对象_cin_，_cout_，_cerr_，_clog_。其中标准输入流对象为_cin_，用于输入；_cout_为标准输出流对象，用于输出。

[1.2] 用标准输入流对象 cin 与提取操作符 >> 连用进行输入时，将_空格_与_回车_分别当作分隔符与结束符。使用成员函数_get() / getline()_进行输入时可以指定输入结束符。

[1.3] 根据文件内容的_数据格式_可分为两类文件：_文本文件_和_二进制文件_。前者和后者存取数据的最小信息单位别为_字节_和_字节_。

[1.4] 文件输入是指从文件向_内存_读入数据；文件输出则指从_内存_向文件输出数据。文件的输入输出首先要_打开文件_；然后_进行读写_；最后关闭文件。

[1.5] 文本文件是存储 ASCII 码字符的文件，文本文件的输入可用 cin 从输入文件流中提取字符实现。文本文件的输出可用_cout_将字符插入到输出文件流来实现。

[1.6] 二进制文件是指直接将计算机内的数据不经转换直接保存在文件中。二进制文件的输入输出分别采用_read()_、_write()_成员函数。这两个成员函数的参数都是_2_个，参数的意义分别是_读写缓冲区_和_字节数_。

[1.7] 设定、返回文件读指针位置的函数分别为_seekg_，_tellg_；设定、返回文件写指针位置的函数分别为_seekp_，_tellp_。

二．选择题

[2.1] 要进行文件的输出，除了包含头文件 iostream 外，还要包含头文件（B）。

A．ifstream　　B．fstream　　C．ostream　　D．cstdio

[2.2] 执行以下程序：

```
char *str;
cin >> str;
cout << str;
```

若输入 abcd　1234↙则输出（D）。

A．abcd　　B．abcd 1234　　C．1234　　D．输出乱码或出错

[2.3] 执行下列程序：

```
char a[200];
cin.getline(a, 200, ' ');
cout << a;
```

若输入 abcd　1234↙则输出（A）。

A．abcd　　B．abcd 1234　　C．1234　　D．输出乱码或出错

[2.4] 定义 char *p = “abcd”，能输出 p 所指向的内存地址的为（D）。

A．cout << &p;　　B．cout << p;

C．cout << (char*)p;　　D．cout << (void *)p;

[2.5] 当使用 ifstream 定义一个文件流，并将一个打开文件的文件与之连接，文件默认的打开方式为（A）。

A．ios::in　　B．ios::out　　C．ios::trunc　　D．ios::binary

[2.6] 从一个文件中读一个字节存于 char c，正确的语句为（B）。

A．file.read(reinterpret_cast<const char *>(&c), sizeof(c));

B．file.read(reinterpret_cast<char *>(&c), sizeof(c));

C．file.read((const char *)(&c), sizeof(c));

D．file.read((char *)c, sizeof(c);

[2.7] 将一个字符 char c = ‘A’写到文件中错误的语句为（D）。

A．file.write(reinterpret_cast<const char *>(&c), sizeof(c));

B．file.write(reinterpret_cast<char *>(&c), sizeof(c));

C．file.write((char *)(&c), sizeof(c));

D．file.write((const char *)c, sizeof(c));

[2.8] 读文件最后一个字节（字符）的语句为（B）。

A．myfile.seekg(1, ios::end);
c = myfile.get();

B．myfile.seekg(-1, ios::end);
c = myfile.get();

C．myfile.seekp(ios::end, 0);
c = myfile.get();

D．myfileseekp(ios::end, 1);
c = myfile.get();

[2.9] read 函数的功能是从输入流中读取（D）。

A.一个字符　B.当前字符　C.一行字符　D.指定若干字节

[2.10] 设已定义浮点型变量 data，以二进制方式把 data 的值写入输出文件流对象 outfile 中去，正确的每句是（C）。

A. outfile.write((double*)&data, sizeof (double));

B. outfile.write((double*)&data, data);

C. outfile.write((char*)&data, sizeof (double));

D. outfile.write((char*)&data, data);

三．简答题

3.1　为什么 cin 输入时,空格和回车无法读入？这时可改用哪些流成员函数？

答：因为空格和回车分别是数据的分隔符和结束符，当输入空格和回车时无法读入。可改用 cin.get()和 cin.getline()等流成员函数。

3.2　在 ios 类中定义的文件打开方式中，公有枚举类型 open_mode 的成员 in、out、app、binary 各代表什么文件打开方式？

答：in：打开文件用于输入操作（从文件读取），若文件不存在则返回失败。

out：打开文件用于输出（写入文件），若文件存在则清空文件。若文件不存在则先创建文件。

app：打开文件用于输出，新数据只能添加在尾部。若文件不存在则先创建文件。

binary：以二进制方式打开文件。

3.3　简述文本文件和二进制文件在存储格式、读写方式等方面的不同，各自的优点。

答：1. 存储格式：文本文件一般用于存储具有“行”结构的文本数据，只包含可显示字符和有限的几个控制字符。二进制文件一般用于存储无显式结构的数据，可以包含任意的二进制字节。

2. 读写方式：文本文件的读写可以使用插入和抽取操作符。二进制文件首先需要 ios:binary 格式来打开，读写时需要使用流成员函数。

3. 优点：使用文本文件保存对象，操作相当简单。使用二进制文件，可以控制字节长度，读写数据时不会出现二义性，可靠性高；同时，由于不知道格式，所以无法读取，保密性好。

3.4　文件的随机访问为什么总是用二进制文件，而不用文本文件？

答：在 C++中可以由程序来控制位置指针的移动，从而实现文件的随机访问，即可读写流中任意一段内容。一般文本文件很难准确定位，所以随机访问多用于二进制文件。

3.5　怎样使用 istream 和 ostream 的成员函数来实现随机访问文件？
答：可以使用 istream 的以下成员函数：seekg(), tellg()。
可以使用 ostream 的以下成员函数：seekp(), tellp()。