



计算机与软件工程学院

上机实验报告

**（ 2019/2020 学年 第 2 学期 ）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程名称 | C/C++程序设计 | | | |
| 课程代码 | 190901019 | | | |
| 任课教师 | 王晓明 | | | |
| 学生姓名 | 刘唐 | 年级 | 19级 |
| 学号 | 3120190971181 | 专业 | 计算机类 |
| 实验成绩 |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 实验7：顺序访问文件 | | | 实验地点 | Home🏢 |
| 实验类型 | 设计 | 实验学时 | 2 | 实验日期 | 2020/6/6 |
| * 撰写注意：版面格式已设置好（不得更改），填入内容即可。 | | | | | |
| 1. 实验目的   掌握顺序读写文本文件、二进制文件。 | | | | | |
| 1. 实验内容 2. 实验任务1   统计ASCII文件内各种字符的数量  要求得到小写单词的数量，大写单词的数量，数字的数量，符号的数量  （就ASCII表里面有的那些。其它编码方式比如汉字，特殊表情还没研究出来怎么区分）  程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   要两个txt文件，名字分别是a.txt以及THE FORCE AWAKENS.txt  //已经在“源文件”文件夹中给出  前者是测试用，随意编辑。  后者是我学英语用的，下载的星球大战之原力觉醒的英语剧本，也可编辑   1. **数据存储**（输入数据在内存中的存储）   一个char型字符c   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   只需要将文件指针内容赋值给c，然后判断其类型，在相应类型统计变量上自增。  利用循环读入指针内容实现整篇文章的统计  利用peek函数找EOF退出文件读取   1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大）      1. 实验任务2     程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   任意20个整数  键盘输入   1. **数据存储**（输入数据在内存中的存储）   整型数组（20个单位）   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   第一题：//其实就是用了一个i变量控制输入【中间那个i==0判断语句】    第二题：  将f1.dat以输入方式打开  将f2.dat以app方式打开  读取f1.dat中的数据到a[]中  将a[]中的数据写入f2.dat中  第三题：  将f2.dat中的数据全部读取到a[]中  冒泡排序（啊这个就不用写流程了吧😄）  以out方式打开f2.dat（默认输出打开，如果已有此名字的文件，则将其原有内容全部清除）  再将a[]中的数据写入f2.dat   1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大）          1. 实验任务3   将ASCII文件转换为二进制文件  程序设计   1. **数据输入**（输入哪些数据、个数、类型、来源、输入方式）   只有一点要求：正常文章就行  （每个字符串不长于30个字符）  在程序提示输入的时候打开source.txt输入。  键盘或者复制粘贴都行   1. **数据存储**（输入数据在内存中的存储）   一个字符串c[30]存储从source.txt文件中读取的单词   1. **数据处理**（说明处理步骤。若不是非常简单，需要绘制流程图）   source.txt以ASCII文件输入打开，targer.txt以二进制文件输出打开  然后如流程图：  +-----------------+  | char c[35] |  +-----------------+  |  V  N /------------------\  +------|infile.peek()!=Eof|<------+  | \------------------/ |  | | Y |  | V |  | +-----------------+ |  | | infile >> c | |  | +-----------------+ |  | | |  | V |  | +-------------------------+ |  | | outfile << c << " " |---+  | +-------------------------+  |  | +-------------------+  +----->| infile.close() |  +-------------------+  |  V  +-------------------+  | outfile.close() |  +-------------------+   1. **数据输出**（贴图：程序运行结果截图。图幅大小适当，不能太大）   **这个记事本打开ASCII文件可以读取，打开二进制文件也一样读取，而且读取出来的文字内容都是一样的。只能从排版上看是否转换了：程序忽略了回车符号，所以所有文字都是“一个单词+一个空格”的输出版式。我这里又复制了一段有排版的文字，然后转换文件变成了上述。放大应该可以看清楚文字~** | | | | | |
| 1. 实验环境 2. 操作系统：Windows10 3. 开发工具：Visual Studio 2019 4. 实验设备：PC | | | | | |
| 1. 提交文档   提交本实验报告（电子版），文件名命名：实验X\_报告\_学号\_姓名.doc  教师将**批阅后（有分数）**的全体学生实验报告刻入一张光盘存档，保证光盘可读。 | | | | | |
| 1. 附：源代码   程序一----------------------------------------------------------  #include<fstream>  #include<iostream>  using namespace std;  int C(ifstream& file);  int main(void)  {  /\*char f[20];  for (int i = 0; (i < 20)&&(i!='\n'); i++)  {  cin >> f[i];  }\*///用`Z搞不定，还是在程序内就找好吧  for (int i = 0; i < 2; i++)  {  if (i==0)  {  ifstream file("a.txt", ios::in);  if (!file)  {  cerr << "open file error" << endl;  exit(1);  }  cout << "a.txt的字数统计如下：" << endl;  C(file);  }  else  {  ifstream file2("THE FORCE AWAKENS.txt", ios::in);  if (!file2)  {  cerr << "open file error" << endl;  exit(1);  }  cout << "THE FORCE AWAKENS.txt的字数统计如下：" << endl;  C(file2);  }  }  }  int C(ifstream &file)  {  char c;  int l = 0, b = 0, d = 0, ch = 0;  //小写字母，大写，数字， 符号  while (file.peek() != EOF)  {  file >> c;  if ((c >= 33 && c <= 47) || (c >= 58 && c <= 64) || (c >= 91 && c <= 96) || (c >= 123))  {  ch++;  }  else if (c >= 48 && c <= 57)  {  d++;  }  else if (c >= 65 && c <= 90)  {  b++;  }  else  {  l++;  }  }  file.close();  cout << "小写字母有" << l << "个" << endl;  cout << "大写字母有" << b << "个" << endl;  cout << "数字有" << d << "个" << endl;  cout << "符号有" << ch << "个" << endl << endl;  return 0;  }  备注：程序一需要两个数据文件：a.txt以及THE FORCE AWAKENS.txt  文件名不能改变，内容可编辑。已经在“源代码”文件夹内“实验一的文件夹内”直接给出  程序二：----------------------------------------------------------------------  #include<fstream>  #include<iostream>  using namespace std;  int main(void)  {  int a[20] = { 0 };  ofstream outfile1("f1.dat", ios::out);  ofstream outfile2("f2.dat", ios::out);  if (!outfile1 || !outfile2)  {  cerr << "open error" << endl;  exit(1);  }  cout << "请输入20个整数：";  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 10; j++)  {  cin >> a[j];  if (i == 0) outfile1 << a[j] << " ";  else outfile2 << a[j] << " ";  }  }  outfile1.close();  outfile2.close();//(1)finish  cout << "已经完成第一小题：将20个整数分别存放在两个磁盘文件中（每个文件中存放10个整数）" << endl << "请打开项目中的f1.dat，f2.dat查看" << endl;  system("pause");  ifstream infile1("f1.dat", ios::in);  ofstream outfile("f2.dat", ios::app);  for (int i = 0; i < 10; i++)  {  infile1 >> a[i];  outfile << a[i] << " ";  }  infile1.close();  outfile.close();//(2)finish  cout << "已经完成第二小题：将f1.dat中的数据存放到f2.dat文件原有数据的后面" << endl << "请打开项目中的f2.bat查看" << endl;  system("pause");  ifstream ifile("f2.dat", ios::in);  for (int i = 0; i < 20; i++)  {  ifile >> a[i];  }  int temp;  for (int i = 19; i >= 0; i--)//冒个泡  {  for (int j = 0; j < i; j++)  {  if (a[j]>a[j+1])  {  temp = a[j];  a[j] = a[j + 1];  a[j + 1] = temp;  }  }  }  ifile.close();  ofstream ofile("f2.dat", ios::out);  for (int i = 0; i < 20; i++ ) ofile << a[i] << " ";  ofile.close();  cout << "已经完成第三小题：将f2.dat中的数据按从小到大的顺序存放到f2.dat（覆盖原来的数据）" << endl << "请打开项目中的f2.bat查看" << endl;  system("pause");  }  程序三-----------------------------------------------------------------------------------：  #include<fstream>  #include<iostream>  using namespace std;  int main()  {  ofstream creat("source.txt", ios::out);  if (!creat)  {  cerr << "creat file failed";  cout << endl;  }  creat.close();//创建这个文件，暂时只能用这个方法，noreplace或者trunc都没用  /\*  fstream creat("source.txt", ios::trunc);  if (!creat)  {  cerr << "creat file failed";  cout << endl;  }  creat.close();  \*///创建文件没用啊不知道怎么回事  cout << "The source file has been created.Please 输入 into \"soruce.txt\" by keyboard or copy." << endl;  cout << "原文件已被创建，请通过键盘或复制粘贴往“source.txt”文件中输入" << endl;  system("pause");  ifstream infile("source.txt",ios::in);//open file by read in  if (!infile)  {  cerr << "Open source file error" << endl;  cout << "请创建原文件（文件名：source.txt)，并在文件内编辑。" << endl;  abort();  }  ofstream outfile("target.txt", ios::binary|ios::out);  if (!outfile)  {  cerr << "Open target file failed." << endl;  abort();  }  char c[35];  while (infile.peek()!=EOF)  {  infile >> c;  outfile << c << " ";  }  infile.close();  outfile.close();  cout << "转换完毕，转换后内容已存入“target.txt”文件" << endl;  system("pause");  } | | | | | |