**中国矿业大学计算机学院实验报告**

|  |
| --- |
| 课程名称  高级语言程序设计           实验名称 输入输出流  班级 计算机科学与技术2019-3班 姓名 王杰永 学号 03190886  仪器组号\_\_\_\_\_\_\_  实验日期 2020.12.17  实验报告要求：1.实验目的   2.实验内容（题目描述，源代码，运行截图，调试情况）    3.实验体会 |
| 一、实验目的  （1）理解C++输入输出流的含义和实现方法。  （2）掌握标准输入输出流的应用，包括格式的输入输出。  （3）掌握对文件的输入输出操作。  二、实验内容  1、第一题  1.1题目描述  【问题描述】  编程序，对 k=1，2，3，…，14，15，按下式分别计算出 15 组（i，d，c）：整数 i=2\*k-1；实数 d=k\*k+k-9.8；字符 c='H'+k。并通过使用运算符“<<”将这15组数据保存到当前目录下自定义的text型磁盘文件ft.txt之中。  【输入形式】      无 【输出形式】      文件内容要求：每组占一行，每行中的输出项用一个空格进行分割。 【样例输入】     无 【样例输出】  文件内容格式如下：  1 -7.8 I  3 -3.8 J  5 2.2 K  7 10.2 L  ....以下省略....  1.2 源代码  #include<iostream>  #include<fstream>  using namespace std;  int main(){      int i;      double d;      char c;      ofstream file("ft.txt",ios::out);      if(!file){          cout<<"文件打开错误！";          return -1;      }      for(int k=1;k<=15;k++){          i=2\*k-1;          d=k\*k+k-9.8;          c='H'+k;          file<<i<<" "<<d<<" "<<c<<endl;      }      file.close();      system("pause");      return 0;  }  1.3 运行截图    1.4 调试情况  在for循环中，应该是for(int k=1;k<=15;k++)，但是最初写的时候写成了i<=15，导致最初的调试错误。  2、第二题  2.1题目描述  【问题描述】  通过使用运算符“>>”和“<<”，对自定义的text型磁盘文件进行如下的指定操作。  （1）对i=1，2，3，...，15，计算d=i\*i+0.5，并将15个结果写出到自定义的text型磁盘文件f1.txt中（注意，f1文件中的数据从小到大有序）；  （2）对i=1，2，3，...，10，计算d=10\*i+0.5，并将10个结果写出到自定义的text型磁盘文件f2.txt中，而后再写出第11个数：357.9（注意，f2文件中的数据也从小到大有序）；  （3）读出文件f1.txt以及f2.txt中的数据，用来形成text型磁盘文件f3.txt，但要求存放于f3.txt中的数据仍然是有序的（按照从小到大的顺序）；  【输入形式】      无 【输出形式】     输出到f3.txt文件中（一行，数据间间隔为一个空格） 【样例输入】      无 【样例输出】      1.5 4.5 9.5 10.5 16.5...（以下略）  2.2 源代码  #include<iostream>  #include<fstream>  using namespace std;  int main(){     fstream file("f1.txt",ios::out);     if(!file){         cout<<"文件打开失败！";         return -1;     }     for(int i=1;i<=15;i++)          file<<double(i)\*i+0.5<<" ";      file.close();      file.open("f2.txt",ios::out);      if(!file){          cout<<"文件打开失败！";          return -1;      }      for(int i=1;i<=10;i++)          file<<10.0\*i+0.5<<" ";      file<<357.9<<" ";      file.close();      double data[26];      file.open("f1.txt",ios::in);      if(!file){          cout<<"文件打开失败！";          return -1;      }      for(int i=0;i<15;i++)          file>>data[i];      file.close();      file.open("f2.txt",ios::in);      if(!file){          cout<<"文件打开失败！";          return -1;      }      for(int i=15;i<26;i++)          file>>data[i];      file.close();        for(int i=0;i<25;i++){          int r=i;          for(int j=i+1;j<26;j++){              if(data[i]>data[j])                  r=j;          }          double tmp=data[i];          data[i]=data[r];          data[r]=tmp;      }      file.open("f3.txt",ios::out);      if(!file){          cout<<"文件打开失败!";          return -1;      }      for(int i=0;i<26;i++)          file<<data[i]<<" ";      file.close();      system("pause");      return 0;  }  2.3 运行截图    2.4 调试情况  在将两个文件的数据合并到一个新的文件且排序时，排序算法使用选择排序，在定义中间变量时，最初使用int型导致最后文件中出现整数。  3、第三题  3.1题目描述  【问题描述】  使用 getline 成员函数读入某个text文件（f5.txt）中的“一篇文章”（如可以是一个C++源程序，文件名由用户从键盘输入），而后将该文件的各空行以及各程序行中的注解均删除掉（认为该行中从第一个双撇符号“//”开始直至行末的所有符号为注解），并将删除注解后的结果行写出到另一个文件（f6.txt）中。  【输入形式】  逐行读入文件f5.txt的内容。  【输出形式】 【样例输入】  文件f5.txt的内容如下：  //This is test  #include <iostream>  int main()  {  cout<<"Hello World"<<endl;  return 0;  }  【样例输出】  文件f6.txt的内容如下：  #include <iostream>  int main()  {  cout<<"Hello World"<<endl;  return 0;  }  3.2 源代码  #include<iostream>  #include<fstream>  using namespace std;  int main()  {      fstream f5("f5.txt",ios::in);      fstream f6("f6.txt",ios::out);      if(!f5&&!f6){          cout<<"文件读取失败!";          return -1;      }      char ch[500];bool flag=false;      while(f5.getline(ch,500)){          flag=false;          char\* p=ch;          while(\*p!='\0'){                if(\*p!='/'){                  f6<<\*p;                  flag=true;              }              else if(\*p=='/'&&\*(p+1)=='/'){                  break;              }              p++;          }          if(flag){              f6<<'\n';          }      }      f5.close();      f6.close();      system("pause");      return 0;  }  3.3 运行截图    3.4 调试情况  起初没有考虑到文件流读到空行该怎么办，最后加入一个标志flag，如果前面已经读取过内容了那么flag设置为true，此时在本行读取完成后会输出一个’\n’。  三、实验体会  本次实验加深了自己对C++输入输出流的理解，特别是掌握了文件输入流对象的getline成员函数的使用方法。 |