**实验报告**

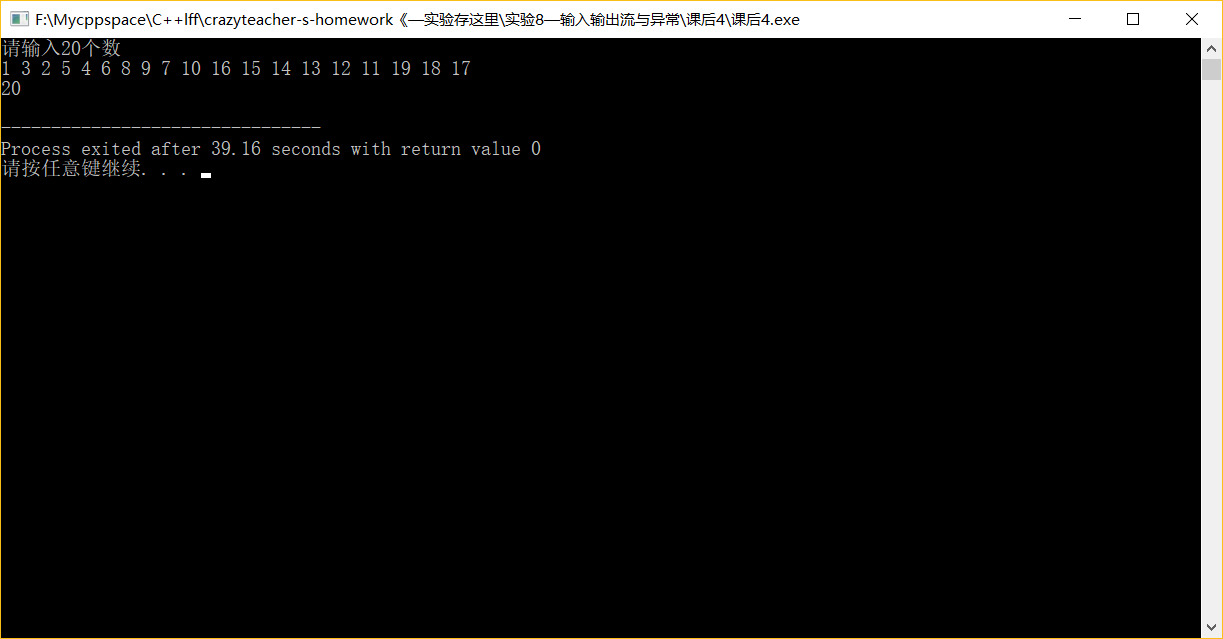
课后题4

实验目的：学会文件读入读出

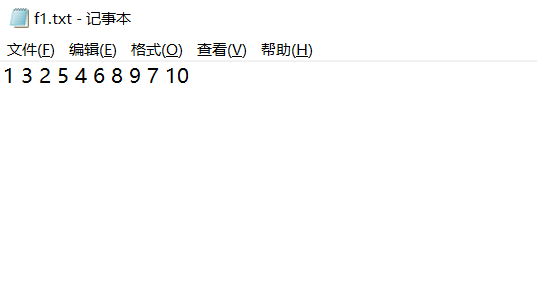
实验代码：

|  |
| --- |
| #include<iostream>  #include <fstream>  #include <algorithm>  using namespace std;  int main()  {  ofstream out("f1.txt",ios::out);  int a[11],b[11];  cout<<"请输入20个数\n";  for(int i=0;i<10;i++)  {  cin>>a[i];  out<<a[i]<<" ";  }  out.close();  out.open("f2.txt",ios::out);  for(int i=0;i<10;i++)  {  cin>>b[i];  out<<b[i]<<" ";  }  out.close();  ifstream in("f1.txt",ios::in);  int c[11];  for(int i=0;i<10;i++)  {  in>>c[i];  }  in.close();  out.open("f2.txt",ios::out|ios::app);  for(int i=0;i<10;i++)  {  out<<c[i]<<' ';  }  out.close();  in.open("f2.txt",ios::in);  int d[22];  for(int i=0;i<20;i++)  {  in>>d[i];  }  in.close();  out.open("f2.txt",ios::out);  sort(d,d+20);  for(int i=0;i<20;i++)  {  out<<d[i]<<" ";  }  } |

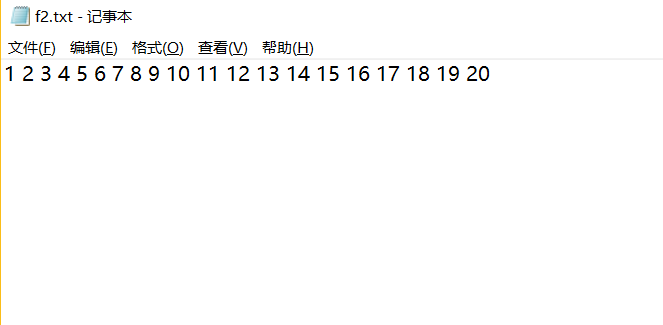
运行结果：



文件f1截图：



文件f2截图：



结果分析：

实现了题目要求的，将二十个数字存入两个文件中，并在排序后输入到文件f2中保存。

课后题5

实验目的：

文件读入读出修改。

实验代码：

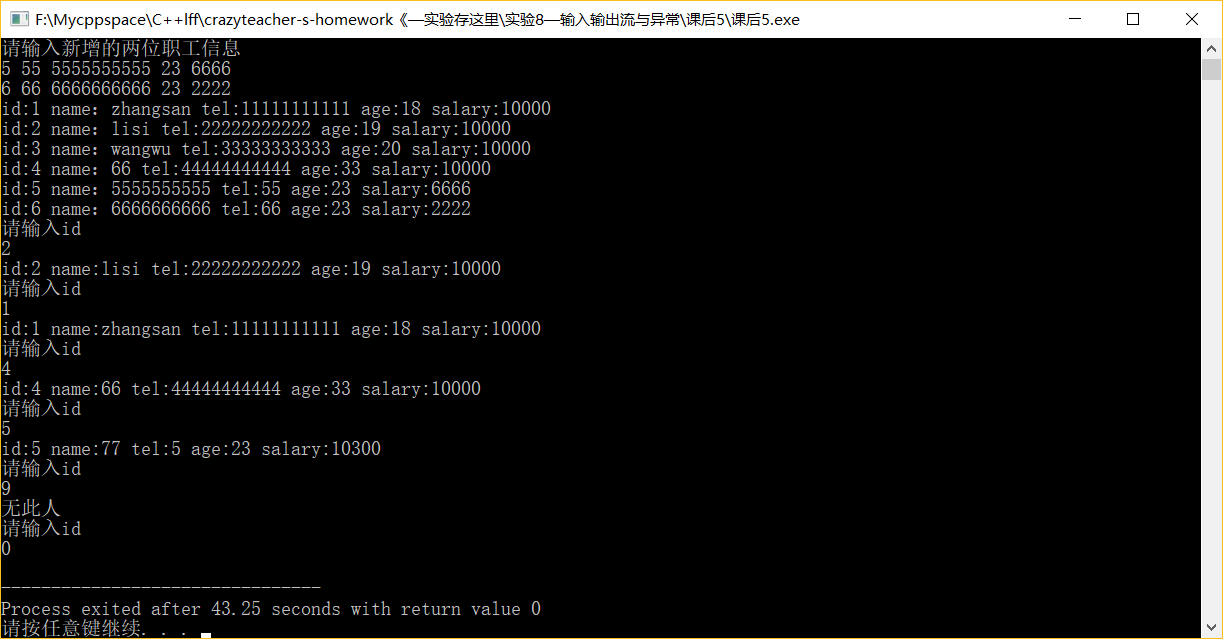
Head

|  |
| --- |
| #include<iostream>  #include <fstream>  using namespace std;  class worker  {  public:  worker(){}  worker(long long i,string t,string n,int a,double s):id(i),tel(t),name(n),age(a),salary(s){}  string tel;  long long id;  string name;  int age;  double salary;  }; |

Main

|  |
| --- |
| #include <iostream>  #include <fstream>  #include "worker.h"  using namespace std;  int main()  {  worker a[100];  a[0] = worker(1,"11111111111","zhangsan",18,10000);  a[1] = worker(2,"22222222222","lisi",19,10000);  a[2] = worker(3,"33333333333","wangwu",20,10000);  a[3] = worker(4,"44444444444","66",33,10000);  a[4] = worker(5,"5","77",23,10300);  ofstream out("职工信息.txt",ios::out);  for(int i=0;i<4;i++)  {  out<<a[i].id<<" "<<a[i].name<<" "<<a[i].tel<<" "<<a[i].age<<" "<<a[i].salary<<endl;  }  out.close();  string name,tel;  int age;  long long id;  double salary;  cout<<"请输入新增的两位职工信息\n";  for(int i=0;i<2;i++)  {  cin>>id>>name>>tel>>age>>salary;  a[5+i] = worker(id,name,tel,age,salary);  }  out.open("职工信息.txt",ios::out|ios::app);  for(int i=0;i<2;i++)  {  out<<a[5+i].id<<" "<<a[5+i].name<<" "<<a[5+i].tel<<" "<<a[5+i].age<<" "<<a[5+i].salary<<endl;  }  out.close();  ifstream in("职工信息.txt",ios::in);  while(in>>id>>name>>tel>>age>>salary)  {  cout<<"id:"<<id<<" name："<<name<<" tel:"<<tel<<" age:"<<age<<" salary:"<<salary<<endl;  }  cout<<"请输入id\n";  int i;  while(cin>>i&&i)  {  for(int i=0;i<7;i++)  {  in>>a[i].id>>a[i].name>>a[i].tel>>a[i].age>>a[i].salary;  }  int flag = 0;  int spot;  for(int j=0;j<7;j++)  {  if(a[j].id == i)  {  spot = j;  flag = 1;  break;  }  }  if(flag)  {  cout<<"id:"<<a[spot].id<<" name:"<<a[spot].name<<" tel:"<<a[spot].tel<<" age:"<<a[spot].age<<" salary:"<<a[spot].salary<<endl;  }  else if(!flag)  {  cerr<<"无此人\n";  }  cout<<"请输入id\n";  }  } |

运行结果：



结果分析：

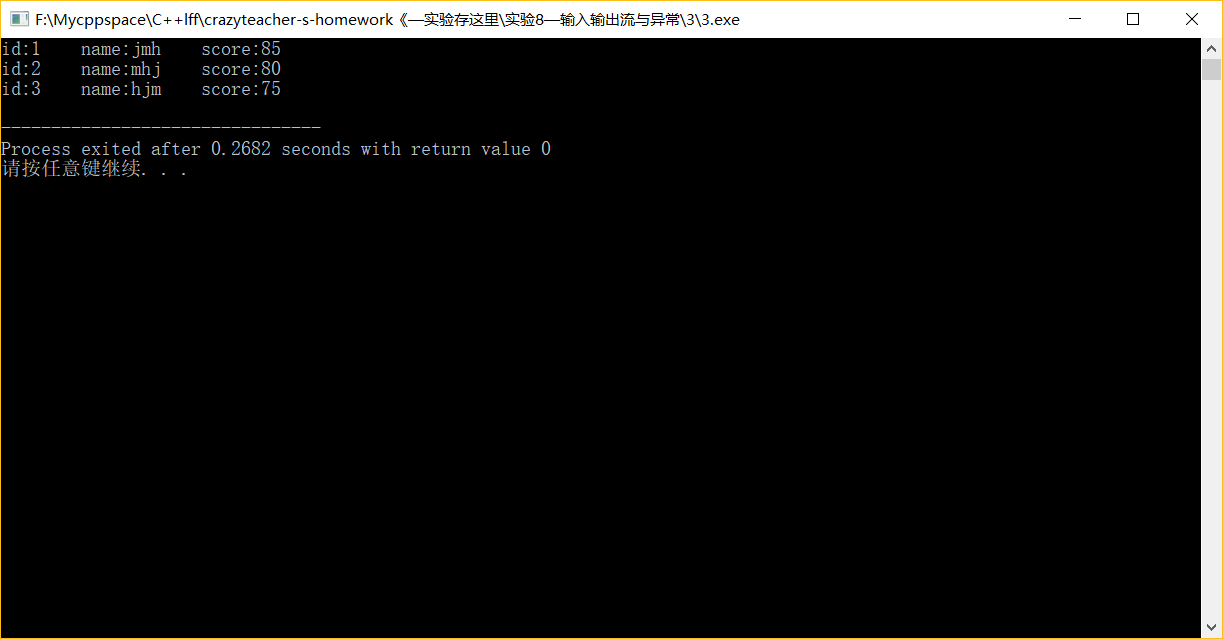
实现了初始录入信息，加入两位新增的信息。循环出入职工号查询信息，0为退出。

1. 设计一个学生类Student，它具有的私有数据成员是：学号、姓名、成绩。建立若干个Student类对象，将它们保存到文件Record.dat中。然后显示文件中的内容。

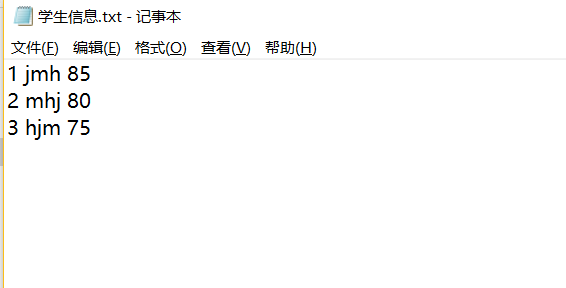
实验目的：熟悉文件输入输出

实验代码：

|  |
| --- |
| #include<iostream>  #include <fstream>  using namespace std;  class student  {  private:  string id;  string name;  double score;  public:  student(){}  student(string i,string n,double s):id(i),name(n),score(s){}  string get\_id()  {  return id;  }  string get\_name()  {  return name;  }  double get\_score()  {  return score;  }  void chuan\_id(string i)  {  id = i;  }  void chuan\_name(string n)  {  name = n;  }  void chuan\_score(double s)  {  score = s;  }  };  int main()  {  student a[100];  a[0] = student("1","jmh",85);  a[1] = student("2","mhj",80);  a[2] = student("3","hjm",75);  ofstream out("学生信息.txt",ios::out);  for(int i=0;i<3;i++)  {  out<<a[i].get\_id()<<' '<<a[i].get\_name()<<' '<<a[i].get\_score()<<endl;  }  out.close();  ifstream in("学生信息.txt",ios::in);  for(int i=0;i<3;i++)  {  string id,name;  double s;  in>>id>>name>>s;  a[i].chuan\_id(id);  a[i].chuan\_name(name);  a[i].chuan\_score(s);  }  for(int i=0;i<3;i++)  {  cout<<"id:"<<a[i].get\_id()<<" name:"<<a[i].get\_name()<<" score:"<<a[i].get\_score()<<endl;  }  } |

运行结果：  


文件中：



结果分析：  
单纯的文件读入读出。

1. 第14章课后题2。

**分析代码执行过程**，**写出运行结果**。运行代码验证答案是否正确。

结果：

Main begin

Call fun（）

Constructor—1101

1101 tan

In get\_data()

Constructor—0

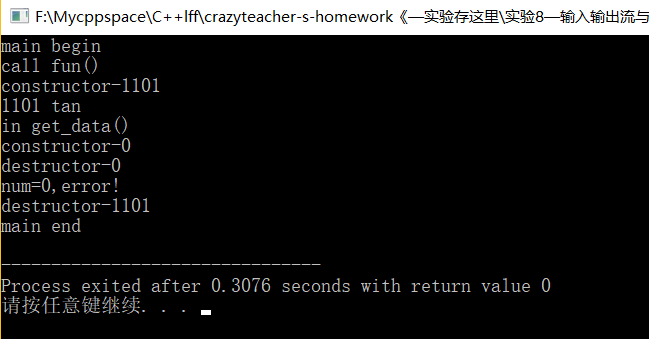
Destructor—0

Num =0,error!

Main end

Destructor—1101

运行结果：



结果分析：

最后两句搞反了，和我的预期不同，错误原因是因为fun函数结束后函数已经析构。再输出最后一句main end.

**5、分析以下代码**，**试写出其运行结果**，并上机验证答案是否正确。

实验目的：了解异常处理的运行机制

实验代码：

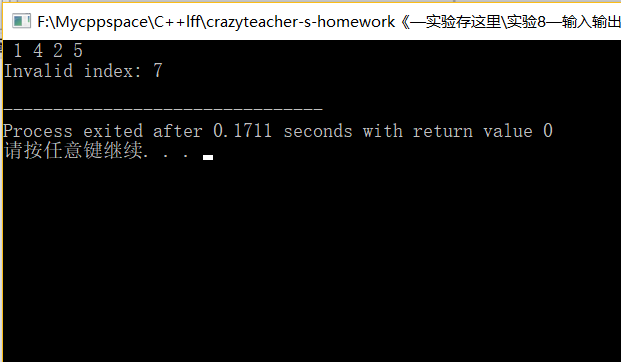
|  |
| --- |
| #include<iostream>  using namespace std;  class FunArray  {  int\* pa, size;  public:  FunArray(int a[], int thesize) : pa(a),size(thesize){ }  int Size(){ return size; }  int& operator[](int index)  {  if(index >= size)  throw index;  return pa[index];  }  };  int main()  {  int s[]={ 3, 5, 2, 0, 7, 4 };  int len = sizeof(s)/sizeof(int);  try  {  FunArray ma(s, len);  ma[3] = 1;  for(int i=0; i<len; i++)  cout<<' '<<ma[s[i]];  }  catch(int err)  {  cout<<"\nInvalid index: "<<err<<"\n";  }  return 0;  } |

预测结果：

1 4 2 5

Invalid index:7

运行结果：



结果分析：

正常输出的1 4 2 5是经过判断符合条件所以正常输出，因为7>6所以属于异常，被安排掉了。

6、富翁换钱计划。有一个富翁，遇到一陌生人。陌生人找他谈了一个换钱的计划。计划如下：我每天给你十万元，而你第一天只需给我一分钱，第二天我仍给你十万元，你给我两分钱，第三天我仍给你十万元，你给我四分钱....，你每天给我的钱是前一天的两倍，直到满一个月（30天）。百万富翁很高兴，欣然接受了这个契约。

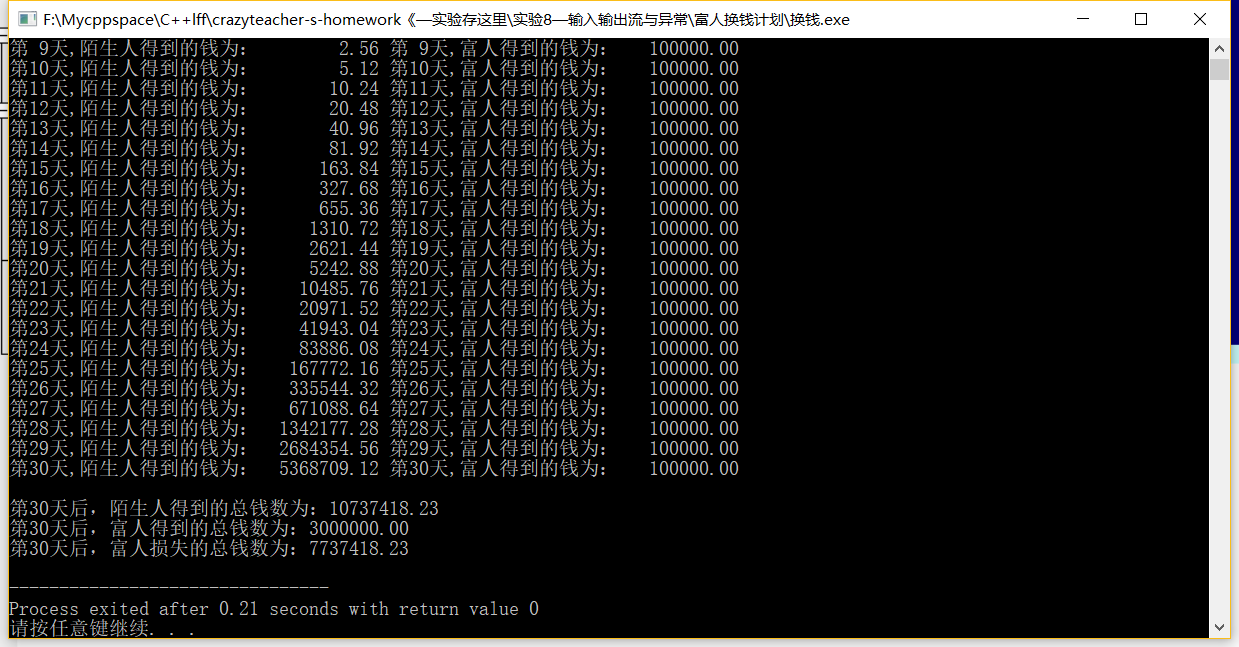
请编写程序，计算一下陌生人最后一共给了富翁多少钱？富翁又给陌生人多少钱？富翁会哭还是会笑呢？

实验目的：我也看不出来目的是啥

实验代码：

|  |
| --- |
| #include <iostream>  #include <iomanip>  #include <cmath>  using namespace std;  int main()  {  double richer = 100000,stranger = 0.01;  double sumrich=0,sumstranger=0;  for(int i=1;i<=30;i++)  {  cout<<"第"<<fixed<<setw(2)<<i<<"天,陌生人得到的钱为：";  cout<<fixed<<setw(12)<<setprecision(2)<<stranger<<" ";  cout<<"第"<<fixed<<setw(2)<<i<<"天,富人得到的钱为：";  cout<<fixed<<setw(12)<<setprecision(2)<<richer<<endl;  sumrich+=richer;  sumstranger += stranger;  stranger \*=2;  }  cout<<endl;  cout<<"第30天后，陌生人得到的总钱数为："<<fixed<<setprecision(2)<<sumstranger<<endl;  cout<<"第30天后，富人得到的总钱数为："<<fixed<<setprecision(2)<<sumrich<<endl;  cout<<"第30天后，富人损失的总钱数为："<<fixed<<setprecision(2)<<sumstranger-sumrich<<endl;  } |

运行结果：



运行结果：一模一样，wow !awesome！好了不闹了，注意下用下iomanip头文件设定下格式就好了。

实验总结：大实验菜单设计真难。