高级语言程序设计 II

实验报告

学号：2190400208

班级：2190400Z

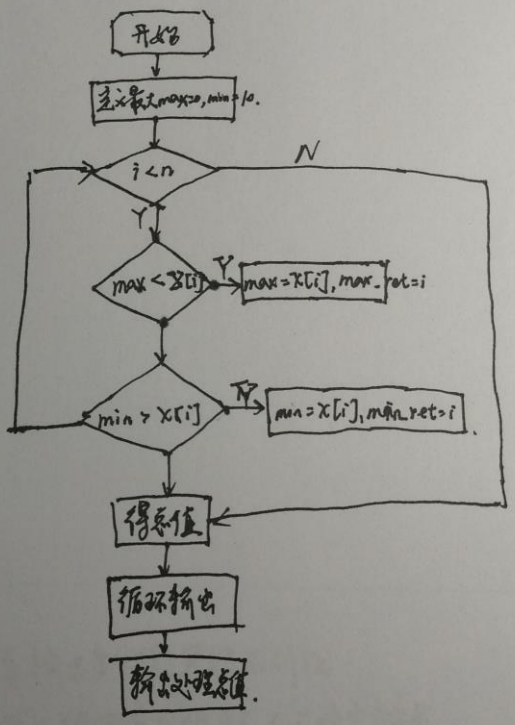
姓名：刘日玉

实验一：简单类　　时间：2024年5月07日　　地点：在线实验室(湖南)

一、实验题目：

某项体育赛事共有N个评委打分，满分为10分，不允许打负分，给某个选手打分后，去除一个最高分和一个最低分，剩余评委分数相加为该选手的本轮得分。用类实现该程序，构造函数规定评委人数N。input函数输入N个评委的打分，output函数输出某选手的本轮得分。

二、数据结构与流程图：



三、核心代码

```
if(maxx<x[i])
{
    maxx=x[i];
    maxx_ret=i;
}
if(minx>x[i])
{
    minx=x[i];
    minx_ret=i;
}
```

```
maxx=0;
minx=10;
```

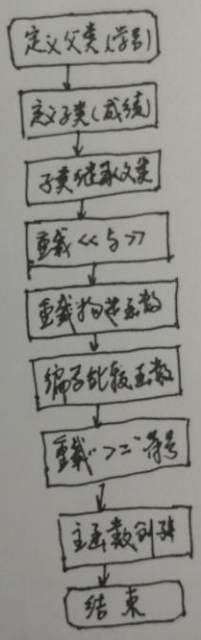
四、总结与收获

在运行结果上没有输出各个值，本题不利于得到最大和最小值，可以使用普通循环，不必非要排序，但是有些现成的排序则尽量使用，后续添加功能会更加方便。

实验二：继承与派生 一 运算符重载 时间：2020年5月13日 地点：在线实验室(湖南)

一、实验题目：学生成绩管理。父类（命名为student1）中有一项学生编号数据number(int型)。派生类（命名为student2）中有一项年终成绩数据score(double)。两个类分别编写构造函数（重载），对数据进行初始化。在子类中对"="运算符重载（使用友元），输入子类对象的数据。对"<<"通过运算符重载（友元），输出子类对象数据。对"<="运算符进行重载。判断子类对象中成绩的高低。主函数中定义了两个子类对象，使用"="输入数据。使用"<="判断两个对象，使用"<<"输出成绩较高的对象的数据。

二、数据结构与流程图：



三、核心代码

```
bool operator <=(Student2 temp)
{
  if(score<=temp.score)
  {
    return true;
  }
  else
  {
    return false;
  }
}
```

四、总结与收获

整体代码可以精简一些，比如核心代码可为 return (score<=temp.score)

为与<<的重载是不一样的，为了方便，<<不用

实验三：利用键盘交互与计时器-容器 时间：2020年5月20日 地点：在线实验(计科楼)

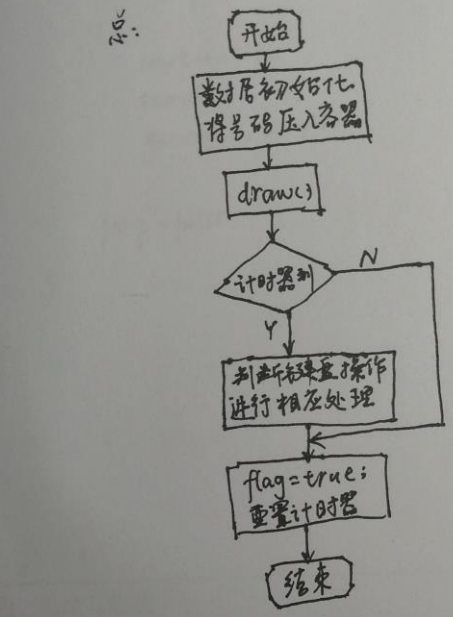
一、实验题目：

模拟一个办事机构(如银行)的叫号程序。

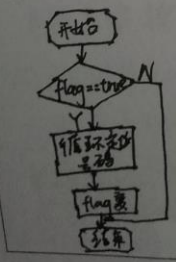
在一个显示区域从上到下按顺序显示5个号码，最开始是1-5，四个方向键控制显示区域的移动。空格键产生一个新号码，将最前面的号码挤出显示区。ESC退出系统。使用键盘交互与计时器交互该程序，使用容器。

二、数据结构与流程图：

总：



draw：



erase：

三、核心代码

```
void draw()
{
    if(flag==true)
    {
        temp1=y;
        gotoxy(x,temp1);
        for(int i=0;i<5;i++)
        {
            cout<<a[i];
            temp1++;
            gotoxy(x,temp1);
        }
        flag=false;
    }
}
```

四、总结与收获

①最开始写清屏函数的时候遇到挫折，只用一个空格符来擦除字符，导致后续擦除不干净。

②事后觉得draw里并不要定义一个新变量，只需最后将y复位即可，既可减少代码量，又增加了可读性。

③擦除使用清屏时系统资源的消耗过大。

实验四：字符串　　时间：2020年5月27日　地点：在线竞赛，湖南

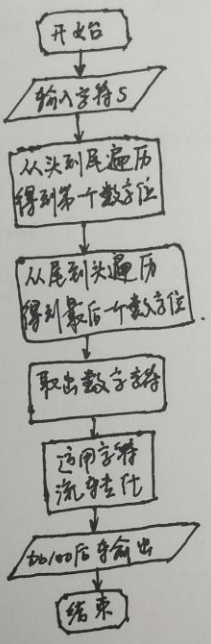
一、实验题目：

从键盘输入一个字符串S，形如：

"abcd345ef"

里面有一个数字字符子串（如上面串中的"345"），将其取出，并且在其数值上+100，输出运算结果。

二、数据结构与流程图：



三、核心代码

```
Void output( )
{
   int ed,beg;
   cout<<"input a string:";
   cin>>s;
   for(int i=0;i<s.size();i++)
   {
      if(s[i]<="9"&&s[i]>='0')
      {
         beg=i; break;
      }
   }
   for(int i=s.size()-1; i>=s.size();i--)
      if(s[i]<='9'&&s[i]>='0')
      { ed=i; break; }
   s1=s.substr(beg,ed-beg+1);
   int n;
   stringstream ss;
   ss<<s1;
   ss>>n;
   cout<<100+n;
}
```

四、总结与收获

①新发现：

s1="345abc" 转为int型可以得到数字345，

但如果s1有字符开头则转化失败.

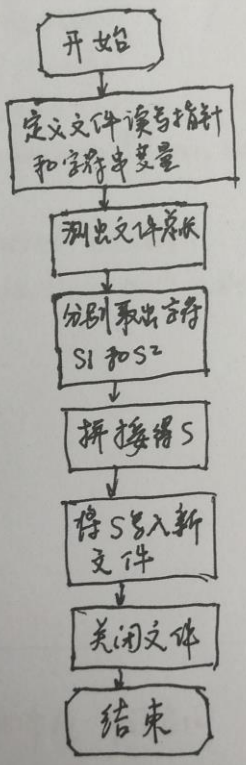
②不足：没有将转化函数分开写，代码不够精简.

实验五：二进制文件　　时间：2020年6月3日　　地点：在线实验（机房）

一、实验题目：

1. 读取文件"exam5.doc"前1500个字符到字符串S1；
2. 读取文件其余内容到字符串S2；
3. 字符串S=S2+S1；
4. 将S串写入"学号姓名.doc"，如"2190400000王小二.doc"
5. 打开生成的文件，有惊喜

二、数据结构与流程图：



三、核心代码

```
void Read()
{
    f2.seekg(0,ios::end);
    len=f2.tellg();
    S.resize(len);
    S1.resize(12500);
    S2.resize(len-12500);
    f2.seekg(0,ios:beg);
    f2.read((char*)S1.c_str(),12500);
    f2.read((char*)S2.c_str(),(len-12500);
    S=S2+S1;
    cout<<"success!"<<endl;
    f1.write((char*)S.c_str(),S.size());
}
```

四、总结与收获

① resize函数的作用：申请空间

② 对文件处理后文件可以由原来的不可打开状态恢复到正常编码.