C++程序设计
实验指导书

2019 年 9 月 24 日

目　　录

第1章　初识C++程序

实验目的

1. 初步学会使用Microsoft Visual Studio集成开发环境（IDE）。
2. 独立上机编写、调试以及运行一个简单的C++程序。
3. 掌握C++程序的基本结构。
4. 掌握C++基本数据类型和运算符。
4. 熟悉输入、输出方法。

实验内容

实验1.1　Visual Studio(VS) 2017项目的设计步骤。

作为学习面向对象的C++的第一步，学习的重点是算法和基本语法，可以建立简单的应用程序进行练习。下面是一个实例，该程序要求用户从键盘输入3个整数，然后按照从小到大的顺序在屏幕上输出。

【步骤一】Visual Studio Community 2017 的下载和安装步骤。

1. 在浏览器网址栏输入：https://visualstudio.microsoft.com或搜索“Visual Studio IDE”，在如图1.1所示选项中选择下载windows版本“Community 2017”，会下载一个小的安装器文件。
2. 双击此安装器文件，进入选择安装页面，在【工作负载】选项中勾选”使用C++的桌面开发“，并设置好安装路径，点击安装即可。如图1.2所示。

【步骤二】启动VS2017集成开发环境。启动并进入VS2017集成开发环境至少有3种方法：

图 1.1　Visual Studio Community 2017下载

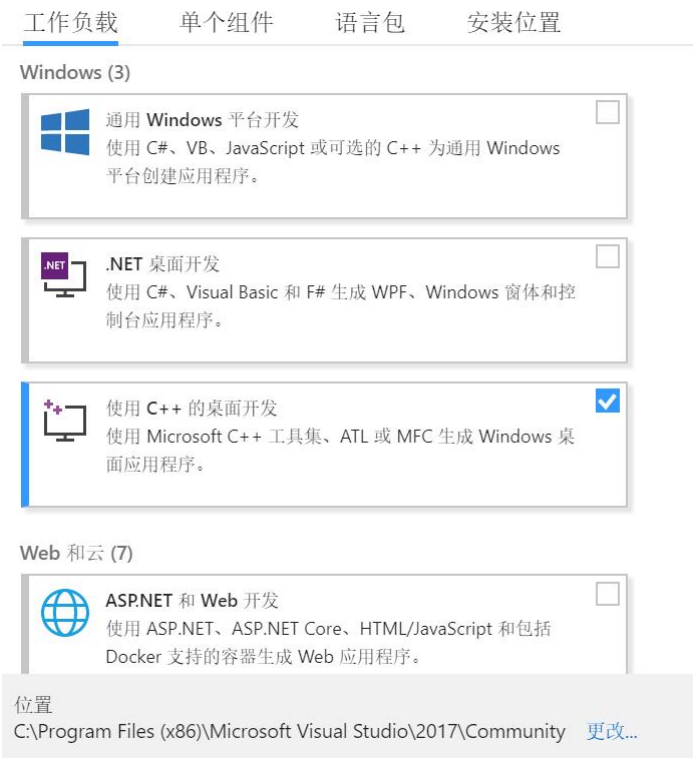

图 1.2　VS2017 C++安装界面

1. 选择开始菜单的“程序”，然后选择Microsoft Visual Studio 2017 级联菜单，再选择Visual Studio 2017菜单项，如图1.3所示。

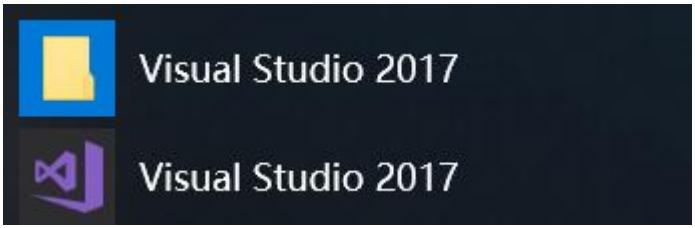

图 1.3　VS2017打开图标

2. 在桌面上创建Microsoft Visual Studio 2017 的快捷方式，直接双击该图标。

3. 如果已经创建了某个C++ 工程，双击该工程的扩展名为.sln的文件图标，也可进入集成开发环境，并打开该工程。选择【文件|退出】菜单，可退出集成开发环境。

【步骤三】创建一个空项目并添加源文件。

1. 进入VS 2017集成开发环境后，选择【文件|新建|项目】菜单项，如图1.4所示，单击【项目】标签，弹出新建对话框，或者选择快捷键Ctrl+Shift+N，进入创建项目页面。 在其

图 1.4　新建一个C++项目

左边的列表框中点击【Visual C++】，然后在右边选项中选择【空项目】，在【名称】文本框中输入项目名exp1_1,在【位置】文本框中输入项目保存的位置，单击【确定】按钮。如图1.5所示，即可创建一个空项目。

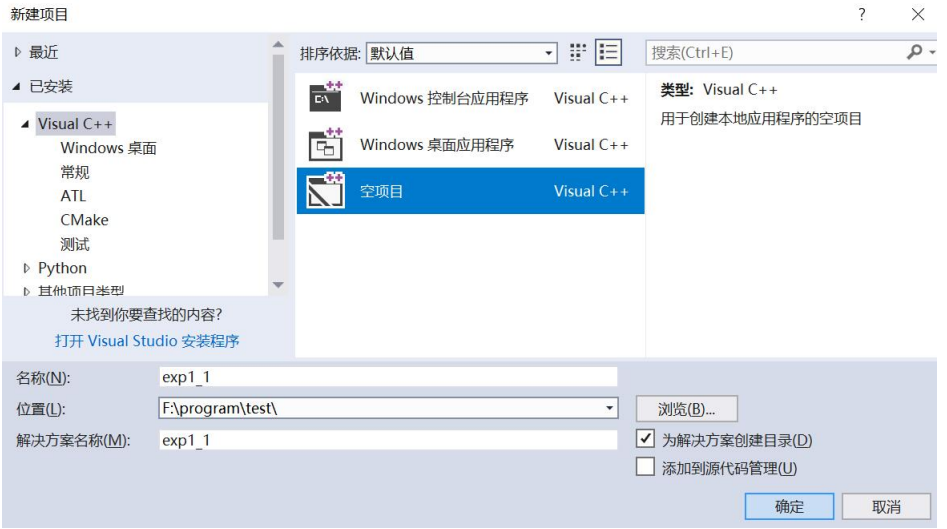

图 1.5　新建项目选项

2. 接下来为空项目新建一个源文件（.cpp），鼠标右键在源文件处单击，选择【添加】|【新建项】，如图1.6所示，或者按快捷键Ctrl+Shift+A也可以。然后在左边的列表框中点击【Visual C++】，在右边选项中选择【C++ 文件(.cpp)】，在【名称】文本框中输入文件名exp1_1，单击【确定】按钮即可。如图1.7所示。

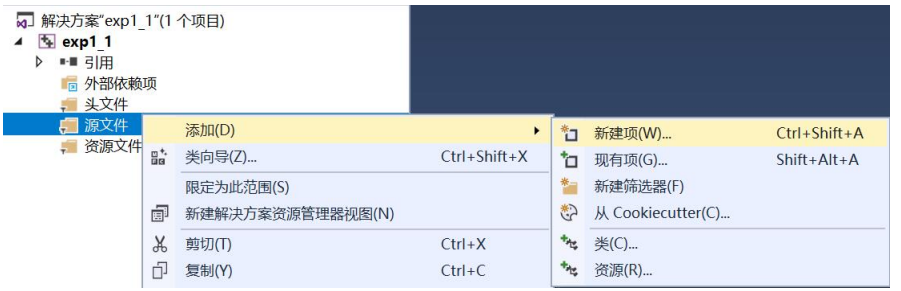

图 1.6　为项目添加文件

【步骤四】程序的编辑、编译、生成和执行。

1. 在新建源文件中编辑源代码，如图1.8所示。也可以添加已经存在的源文件，右击【源文件文件|添加|现有项】菜单项或按组合键Alt+Shift+A，如图1.9所示。在随后打开的插入文件对话框中选择待添加文件，单击【添加】按钮即可将其添加到项目。

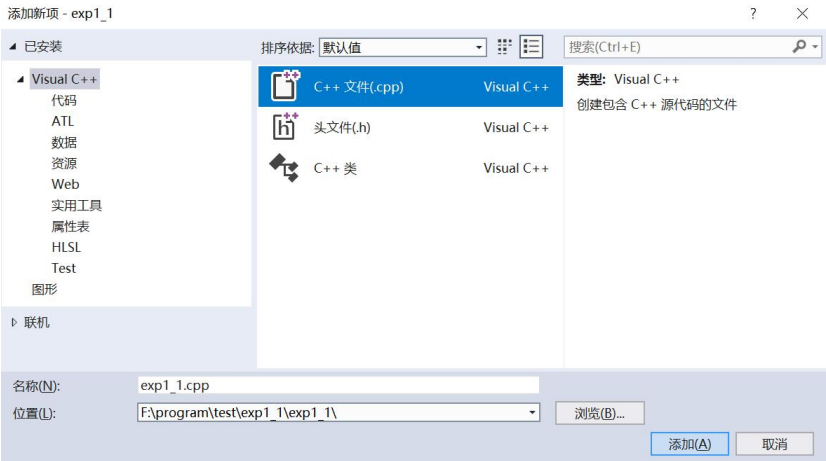

图 1.7　文件选择与命名

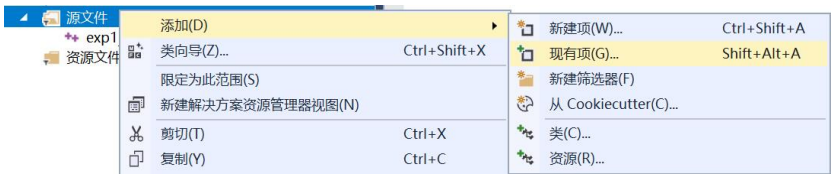

图 1.8　为项目添加已有文件

```
6   int main() {
7       int num1, num2, num3;
8       int max, min;
9       cout << "Please enter 3 integers:\n";
10      cin >> num1 >> num2 >> num3;
11      if (num1>num2) {
12          max = num1;
13          min = num2;
14      }
15      else; {
16          max = num2;
17          min = num1;
18      }
19      if (num3>max) {
20          cout << num3 << '\t' << max << '\t' << min << '\n';
21      }
22      else if (num3<min) {
23          cout << max << '\t' << min << '\t' << num3 << '\n';
24      }
```

图 1.9　源代码编辑

2. 要求输入三个不等的整数后按Enter键，屏幕上由大到小输出这三个整数，如代码清单1.1
【程序】

代码清单 1.1　*

```
#include <iostream>
using namespace std;
int main() {    //程序从main函数开始执行
    int num1, num2,num3;
    int max, min;
    cout << "Please enter 3 integers:\n";
    cin >> num1 >> num2 >> num3;
    if (num1>num2) {
        max = num1;
        min = num2;
    }
    else {
        max = num2;
        min = num1;
    }
    if (num3>max){
        cout << num3 << '\t' << max << '\t' << min << '\n';
     }
    else if (num3<min){
        cout << max << '\t' << min << '\t' << num3 << '\n';
     }
    else{
        cout << max << '\t' << num3 << '\t' << min << '\n';
     }
    return 0; //返回一个整数值
}
```

3. 源代码输入完后，点击菜单栏【生成】|【生成解决方案】或按F7快捷键，即可编译源文件exp1_1.cpp。系统会在输出窗口显示出错（error）信息以及警告（warning）信息，如图1.10所示。编译器在“输出”窗口给出语法错误和编译错误信息。语法错误处理：鼠标双击错误信息可跳转到错误源代码位置处，然后进行修改，一个语法错误可能引发系

统给出很多条error信息。因此，发现一个错误并修改后最好重新编译一次，以便提高工作效率。当错误为0时，可以得到目标文件（exp1_1.obj）。警告错误处理: 一般是触发了C/C++的自动规则，如将一个单精度（浮点）型数据赋给整型变量，需要系统将单精度型数据自动转换为整型，此时小数部分会丢失，因而系统给出警告信息，警告信息不会影响程序执行，本例可以通过强制转换去掉警告信息。

图 1.10　代码调试错误窗口

【步骤四】程序的调试。

程序成功编译之后并不意味着程序能够成功运行，此时还需借助调试工具进行跟踪调试，解决程序潜在的逻辑错误。下面介绍调试过程。

1. 首先，在源程序中可能出现错误的行上设置断点，方法是将光标移至该行，然后按F9 键（再按一次F9键将取消断点），此时该行左侧出现一个红色圆点，断点设置成功。如图1.11就是在A行设置了断点。

2. 然后选择【调试】|【开始调试】菜单命令（也可直接单击工具栏上的图标）或按F5快捷键，程序开始执行。但执行到断点处停止，桌面会跳出自动窗口，自动窗口显示变量的值，若要观察某些变量的值分析并查找出错原因，可以将变量加入监视窗口。如在监视窗口加入max 和min 两个变量进行监视，方法是：在局部变量窗口选择变量max和min，右键单击，选择【添加监视】即可。监视窗口的每一行显示一个对象，其中左栏显示变量名，双击它可进行编辑；中间栏显示变量的值，右栏是变量类型。接下来可按F10 键（不跟踪进入函数内部）或F11 键（跟踪进入函数内部）从断点位置处单步执行。通过单步执行（一条一条语句执行）可以看出我们所观察的变量的变化以及程序执行流程是否正确，如果不正确是由那条语句引起的，因而有针对性的检查错误原因。例如在本例中，尽管num1 > num2, 在执行了if后面的｛max=num1; min=num2; ｝后，通过单步执行发现仍然执行了else 后面的｛max=num2; min=num1; ｝所以当程序执行到箭头所指位置时，max=45, min=48, 如图1.12所示。 与预期结果不相符，说明程序的流程有问题。此时再仔细分析源程序，发现问题在于else 后多余的分号。

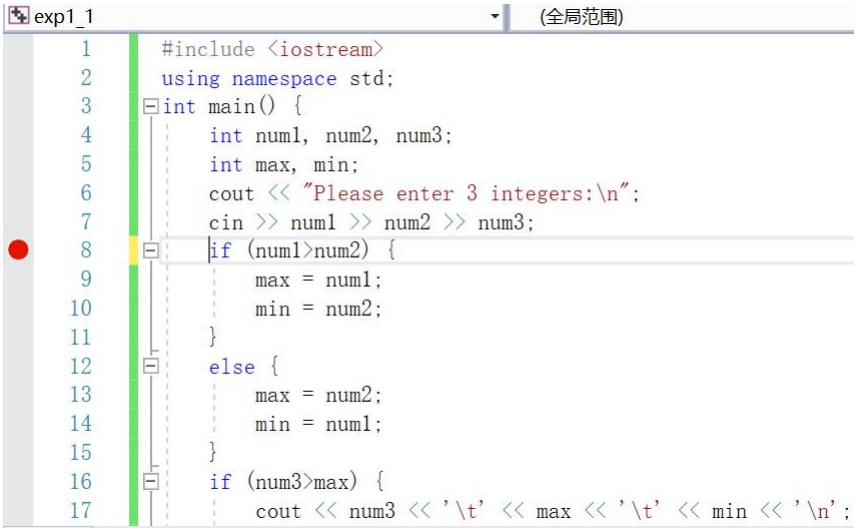

图 1.11　设置断点

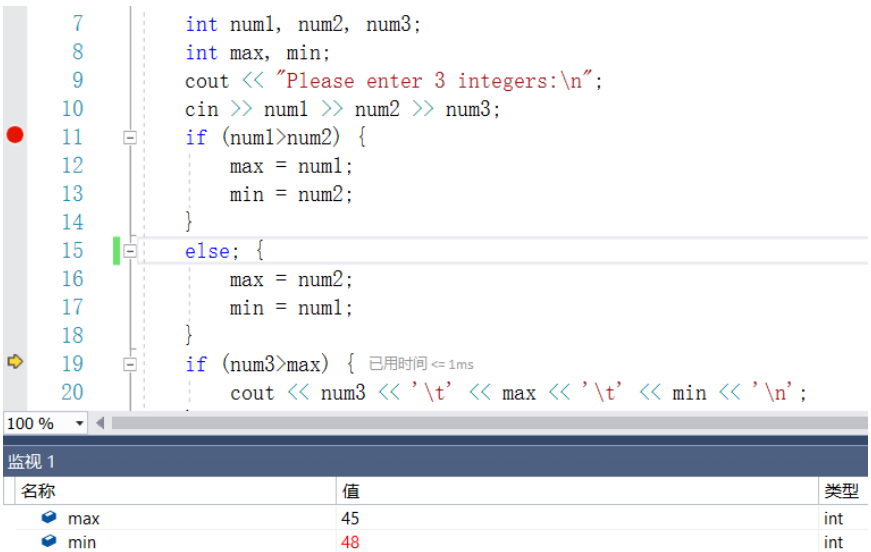

图 1.12　断点调试

调试过程中，Variables 窗口动态显示各变量值随程序执行而变化的结果。在面向对象程序设计中，若程序中有类的对象，Variables 窗口的this 页可显示当前this 指针所指向对象的各个值。经过反复的修改和调试，使程序中所有问题得到改正后，可得到正确的执行结果。如图1.13所示，输入三个整数，按从小到大的顺序输出了三个数。

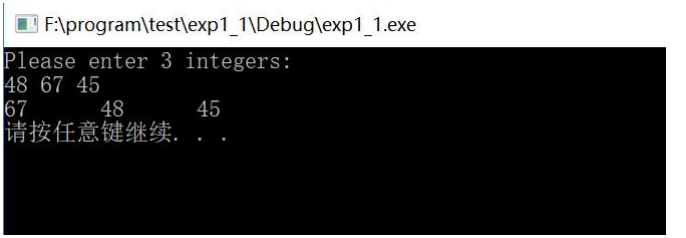

图 1.13　命令窗口结果显示

【练习】

建立简单控制台程序。使用VS2017开发环境来调试以下源程序。

【程序】

代码清单 1.2　*

```
#include <iostream>
#include <cmath>       //sqrt()的头文件
using namespace std;
int main() {
    double num1, num2, num3, s, area;
    cout << "num1,num2,num3 = ";    //输入三角形的三条边
    cin >> num1 >> num2 >> num3;
    s = (num1 + num2 + num3) / 2.0;
    area = sqrt(s*(s - num1)*(s - num2)*(s - num3));//求三角形面积
    cout << "area= " << area << endl;
    return 0;
}
```

【要求】

1. 根据操作过程填写以下表格。

内容	操作	说明或结果分析
进入VS2017		
建立工程为"exp2_2"		
添加源文件		
编辑代码		
编译程序		
运行程序		

2. 采用以下各组数据输入测试，记录输出结果。分析原因，思考如何解决？

3　4　5

3　4　12

0　6　2

-2　7　9

3. 修改程序。

(a) 把double 改为int，重新编译程序，会出现什么编译信息？什么原因？

(b) 把s和area定义为double可以消除编译错误吗？为什么？

(c) 采用以下数据输入测试，记录输出结果，分析原因。

3.45　5.618　4.012

(d) 增加输出a、b、c对象值的语句，观察不同输入时，对象值的变化。

实验1.2　一个简单的C++程序

编写满足下列要求的程序：根据输入的球半径，分别计算球的表面积、体积。球的表面积计算公式为$s=4\pi r^2$,球体积计算公式为$v=\frac{4}{3}\pi r^3$。

【要求】

1. 要求输入前应有提示性输出，如“Please input the radius of the ball ”；
2. 变量的命名，最好能见文知义，如radius, volume ,weight 等；
3. 在计算公式中使用正确的变量数据类型。

第2章　基本数据类型和表达式

实验目的

1. 掌握基本数据类型的内存结构。
2. 掌握运算符的基本属性。
3. 掌握表达式求值的基本方法。

实验内容

实验2.1　基本数据类型

编写程序，其中包括常量，类型推导，类型转换，常用运算符和转义字符的使用。

【要求】

1. 在调试过程中，通过监视窗口观察相关对象值的变化情况；
2. 观察sizeof的输出结果，说明其含义。

【程序】

代码清单 2.1　*

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int r(0);
```

```
5       const double pi = 3.1415926;
6       auto area = 0., sum = 0.;
7       area = pi * r*r;
8       sum += area;
9       cout << "The area is" << area << "\n The sum area is \t "
10           << sum << endl;
11      ++r;
12      area = pi * static_cast<double>(r*r);
13      sum += area;
14      cout << "The area is" << area << "\n The sum area is \t "
15           << sum << endl;
16      cout << "The size of r " << sizeof(r) << " "
17           << "The size of int " << sizeof(int) << endl;
18      cout << "The size of area " << sizeof(area) << " "
19           << "The size of double " << sizeof(double) << endl;
20  }
```

第3章　语句控制结构

实验目的

1.流程控制语句用于实现基本程序结构，是程序设计的基础，要求掌握条件语句和开关语句的使用。

2.掌握3种循环结构：while ,do—while ,for 的区别与联系，以及它们之间相互转换的方法，并能正确使用它们。

3.掌握与循环语句相关的break 语句和continue 语句的使用方法。

实验内容

实验3.1　判断一个数的奇偶性

输入一个数，判断它的奇偶性后输出结果。

【分析】

判断一个数是否为偶数，只需要判断它是否能被2整除，若能，则为偶数，否则为奇数。

【程序】

代码清单 3.1　*

```
#include <iostream>
using namespace std;
int main() {
    int input;
    cout << "which number do you want to test:\n";
    cin >> input;
```

```
7       if (input % 2){
8         cout << "number" << input << '\t' << "is odd.";
9         }
10      else{
11        cout << "number" << input << '\t' << "is even.";
12      }
13      return 0;
14  }
```

【要求】

1. 分别使用数据345 680 -34 作为输入数据，测试程序，分析程序结果并记录。

输入	结果

2. 使用上述数据测试结果正确吗？如果有不正确之处请修改。
3. 请修改if (input%2) 中的表达式，使程序仍然能正确执行。

实验3.2　求一元二次方程的根

编程求一元二次方程$ax^2+bx+c=0$的根。包括以下判断和结果给出提示；$\Delta= b^2-4ac$。若输入a=0，输出一个实根；若Δ>0,输出两个不相等的实根；若Δ=0,输出两个相等实根；若Δ<0，输出两个复数根。

【要求】

1. 分别利用嵌套的if-else 和if-else if -else 结构编写源程序并调试运行，并记录结果。比较两者的不同之处。

输入	结果
a=0 b=0 c=4	
a=0 b=2 c=4	
a=1 b=2 c=0	
a=2 b=5 c=1	

2. 思考if - else 嵌套使用时的注意事项是什么？

实验3.3　根据分数求等级

输入一门课程的成绩，若高于90分，输出“grade A”；若高于80分而低于90分，输出“grade B”；若高于70分而低于80分，输出“grade C”；若高于60分而低于70分，输出“grade D”；否则输出“Failed”。

【要求】

1. 使用if - else 语句和switch 语句两种方法实现。
2. 分析if - else 语句和switch 语句的区别，switch 语句特别适合于什么情况使用？
3. 思考使用switch 语句时应注意什么？

实验3.4　判断一个数是否是3或7的倍数

输入一个数，判断其是否是3或7的倍数，可分为4种情况输出。

【要求】

1. 是3的倍数，但不是7的倍数。
2. 不是3的倍数，是7的倍数。
3. 是3的倍数，也是7的倍数。
4. 既不是3的被数，也不是7的倍数。

实验3.5　大小写字母转换

编写一个程序：从键盘键入一串字符，要求将其实现大小写字母的相互转换，并输出。

【分析】

由ASCII码表可知，大写英文字母的ASCII码值在65至90之间，小写英文字母的ASCII码值在97至122之间，每一个英文字母的大写和小写的ASCII码相差32。

实验3.6　计算求π的近似值

编程计算π的近似值(要求小数点后面10位)。提示:可利用下列公式计算π值：

$\frac{\pi}{2} = 1 + \frac{1}{3} + \frac{1}{3} \times \frac{2}{5} + \frac{1}{3} \times \frac{2}{5} \times \frac{3}{7} + \cdots\cdots$ 第n项为a_n与前一项a_{n-1}的关系是$a_n = a_{n-1} \times \frac{n-1}{2n-1}$

【程序】

代码清单 3.2　*

```
#include <iostream>
using namespace std;
int main() {
    long int i(0);
    double sum(0), pi, term(1.);
    do {
        sum += term;
        i++;
        term = term*(i - 1) / (2 * i - 1.0);
    } while (term >= 1.0e-10);
    pi = 2 * sum;
    cout << "pi=" << pi << endl;
    return 0;
}
```

【要求】

1. 输入程序编译后使用Debug跟踪，单步执行程序，记录以下变量值的变化：

i	term	sum

2. 从跟踪结果分析，term和sum的值有什么错误？
3. 循环结束后，i的值是多少？用什么简单的办法可以看到？
4. 对程序作何修改可以使其得到正确的结果？

实验3.8　循环打印上三角形

打印输出上三角形，如下图所示

```
*
***
*****
*******
*********
```

【分析】

三角形的形状由空白和星组成，分析每一行要输出几个空格和几个星，就得到三角形。

实验3.9　对字符进行统计

输入若干字符，统计其中数字字符、空白字符和其他字符的个数，输入EOF结束。

【分析】

要对字符进行统计，需要循环语句反复输入字符，读入字符后用switch 语句判断字符的种类：要统计3 种字符数，需要定义3个用于计数的变量nDigit ,nWhite ,nOther,并置初始值0；读入字符用cin.get()函数，在这里用while循环比较合适。白字符指空白键、Tab键和回车键。EOF表示End of File , 其值为-1，从键盘输入CTRL+Z键即可。这里采用重载的int cin.get( ) 函数，它返回的是整型数，所以能返回EOF。

【要求】

1. 程序中的变量c，可以定义为char 类型吗？请试之，并解释原因。
2. 请注意程序中case 分支语句后的break语句，break能去掉吗？为什么？
3. 如果要分别统计0-9中个数字出现的次数，怎样才能有效地实现，请修改程序。
4. 如果要统计输入的一段文字中出现的行数、单词数和字符数，又怎样有效地实现，请完成之。

【程序】

代码清单 3.3　*

```
1 #include <iostream>
2 #include <stdlib.h>
3 using namespace std;
```

```
int main() {
  int c;
  int nWhite, nOther, nDigit;
  nWhite = nOther = nDigit = 0;
  c = cin.get();//从流中读取一个字符并返回一个整数
  while (c != EOF) {
    switch (c) {
    case '0':case '1':case '2':case '3':case '4':
    case '5':case '6':case '7':case '8':case '9':
      nDigit++;
      break;
    case ' ':case '\n':case '\t':
      nWhite++;
      break;
   default:
      nOther++;
      break;
    }
    c=cin.get();//读入下一个字符
  }
  cout << "Digits=" << '\t' << nDigit << '\n';
  cout << "White space=" << '\t' << nWhite << '\n';
  cout << "Other Chars=" << '\t' << nOther << '\n';
  system("pause");
  return 0;
}
```

第4章　复合类型、string和vector

实验目的

1. 掌握引用和指针的概念及应用。
2. 熟练应用数组与多维数组。
3. 指针与数组的相互关系。

实验内容

实验4.1　十进制转成二进制

编写程序输入一个十进制表示的正整数，将其转换成二进制并输出结果。方法：除二取余，然后倒序排列，高位补零。

【程序】

代码清单 4.1　*

```
#include<iostream>
int main()
{
    int num, i, j = 0;
    int a[100];//存储二进制编码
    std::cin >> num;
```

```
    i = num;
    while (i)//对2取余并除2，直到商为0时为止
    {
        a[j] = i % 2;
        i /= 2;
        ++j;
    }
    for (i = j - 1; i >= 0; i--)//逆序输出
        std::cout << a[i];
    return 0;
}
```

实验4.2　指针访问数组

从键盘输入十个整数存入一维数组中再按反序输出。

【程序】

代码清单 4.2　*

```
#include<iostream>
using namespace std;
int main() {
    int value[10];
    for (int i = 1; i <= 10; i++) {
        cout << " 请输入第 " << i << " 个整数: " << endl;
        cin >> value[i-1];
    }
    cout << " 反序输出: " << endl;
    for (int i = 9; i >=0; i--) {
        cout << value[i] <<' ';
    }
    return 0;
}
```

【要求】

修改程序，用指针访问数组元素的方法反序输出。

实验4.3　用指针给二维向量赋值

定义一个5×5 二维向量。

【要求】

1. 用指针按照从左往右从上到下的次序依次给向量赋值，值从1开始递增。
2. 用指针将向量的右上部分全部置0（对角线不变）。

实验4.4　判断字符串是否是“回文”

判断用户输入的C字符串是否为"回文"，所谓"回文"是指顺读和反读都一样的串，例如串12321、madam。

【程序】

代码清单 4.3　*

```
#include <iostream>
#include <cstring>
using namespace std;
const int SIZE = 100;
int main(){
    char carray[SIZE];
    int i,len,is_palindrome = 1;
    cout<<"Please input a string ..\n";
    cin.get(carray,SIZE);
    len = strlen(carray);
    for(i=0;i<len/2;i++){
        if(carray[i]!=carray[len-1-i]){
            is_palindrome=0;
            break;}
    }
    if(is_palindrome) cout<<"The string is a palindrome\n";
    else cout<<"The string isn't a palindrome\n";
    return 0;
}
```

【要求】

1. 重新定义回文为：滤去所有非字母字符（包括空格）后，不考虑字母的大小写，从左向右和从右向左读都相同的词或短语。如,”Madam, I’ m adam”和“Golf ,No Sir ,prefer prison flog!”
2. 改写上面程序，用string来代替字符数组，并用指针来完成相同操作。

实验4.5　约瑟夫问题

约瑟夫（Josephus）问题：n 个人围坐成一圈，从1 开始顺序编号；游戏开始，从第一个人开始由1 到m 循环报数，报到m 的人退出圈外，问最后留下的那个人原来的序号。

【分析】

本题可定义一个容器(vector <bool>)，初始化大小（元素个数）为n。容器里元素的值标识该人是否出局，1在圈内，0出局。值为0的元素不参加报数。可用一个整型数k做计数器，采用倒计数，记录留下的人数。

【提示】

容器里的元素是线性排列的，而人是围成圈的，用容器表示要有一种从容器尾部跳到其头部的技巧，即下标加1除以n求余数。

实验4.6　猜字游戏

猜字游戏，规则如下，程序初始化一个单词集合，每次游戏开始时，程序随机选择一个单词让玩家来猜，玩家每次只能猜一个字母，如果选择的单词里有玩家所猜的字母，则玩家猜测成功，程序将显示单词里所有猜中的字母，如果单词里没有所猜字母，则猜测失败，共有六次机会。

【提示】

假如系统选择了单词good，玩家第一次猜测为a，程序提醒玩家猜错，还有5次机会，并显示玩家已经猜过的字母和到当前为止所猜的单词中正确的猜测（没有猜中的字母用*代替，第一次没有猜中所以显示****）；第二次如果玩家猜o，猜测正确，显示玩家到当前为止所猜的单词中正确的猜测，其他未猜中的用*代替（*oo*）；第三次如果玩家猜a，提醒玩家此字母已经猜过，请重试，并显示玩家所有猜错的字母。依次进行，直到玩家猜中或者猜错的次数达到最大限制次数，此次猜单词游戏结束，询问玩家是否继续下一个单词的猜测。

【程序】

代码清单 4.4 *

```
#include <iostream>
#include<string>
#include<cstdlib>  //for srand(),rand()
#include<ctime>    //for time()
using namespace std;
int main() {
    const int NUM = 26;
   //初始化一个单词库
    const string wordlist[NUM] = {
        "program","cat","cereal","danger","good","florid",
        "garage","heal","insult","joke","keeper","loaner",
        "nonce","onset","ok","quilt","remote","stolen","train",
        "useful","valid","where","xenon","cool","result"
   };
    srand(time(0));
    char play;
    cout << "Will you play a word game?<y/n>";
    cin >> play;
    while (play == 'y' || play == 'Y') {
        //从单词库随机选一个单词rand() 产生一个随机正整数
        string target = wordlist[rand()%NUM];
        //*****请在下面补充你的代码*****************//

        //*****代码填充截止************************//
        cout << "Will you play another?<y/n>";
        cin >> play;
   }
    cout << "Bye." << endl;
    return 0;
}
```

第5章　函数

实验目的

1. 掌握函数的定义及调用方法。
2. 掌握函数参数的传递（传值）、形参与实参的关系以及函数声明。
3. 掌握引用作为函数参数的方法。
4. 掌握指针或数组作为函数参数的函数定义及调用方法。
5. 了解内联函数、重载函数、带默认参数函数的定义及使用方法。
6. 掌握作用域的概念、变量的存储类型及它们之间的差别。
7. 掌握程序的多文件组织。

实验内容

实验5.1　用迭代法求平方根的函数

编写一个通用的求平方根的函数，参数（形参）为待求平方根的数，返回值为该数的平方根。由于平方数不能为负数，因此在主调函数中，需要判断输入数的正、负，为正则用该参数（实参）调用求平方根函数；为负则输出错误信息。

【提示】

正数a的平方根迭代公式如下，平方根函数声明为：double sroot(double val)。

$$x_{n+1} = (x_n + \frac{a}{x_n})/2, x_1 = a/2$$

实验5.2　全局变量、局部变量和静态局部变量的应用

分析并写出下列程序的执行结果，然后输入计算机执行，比较分析结果与执行结果。如果两结果不相同请分析原因。

【程序】

代码清单 5.1　*

```
#include<iostream>
using namespace std;
int value1 = 300, value2 = 400, value3 = 500;
void funa(int value3) {
    static int value1 = 5;
    value1 += value3;
    cout << value1 << ' ' << value3 << '\n';
}
void funb(int value1) {
    value1 = value2;
    cout << value1 << '\n';
}
void func() {
    int value3=0;
    cout << value1 <<' '<< value2 <<' '<< value3 << '\n';
    ::value3 -= 100;
}
int main() {
    funa(value1);
    funb(value2);
    funa(value2);
    func();
    cout << value1 << ' ' << value2 << ' ' << value3 << '\n';
    return 0;
}
```

实验5.3　字符串简单的“加密”和“解密”

按一定的规则可以将一个字符串经加密转换为一个新的串，例如加密的简单方法是当为’a’～’y’的小写字母时用后一个字母代替前一个字母，其中’z’变换为’a’，其他字符时不变。
例如：原串为This is a secret code!
加密后的串为Tijt jt b tfdsfu dpef!

编写一个程序对输入串加密，输出加密前和加密后的串，再将加密后的字符串解密输出。主函数如下，请编写加密函数和解密函数。
【程序】

代码清单 5.2　*

```
#include<iostream>
using namespace std;
void secret(char* data);
void desecret(char data[]);//传递数组参数的两种形式
int main(){
    char st[]="This is a secret code!";
    cout<<st<<endl;
    secret(st);
    cout<<st<<endl;
    desecret(st);
    cout<<st<<endl;
    return 0;
}
```

【要求】

1. 将secret(char* data)改为secret(const char* data)，程序还能运行吗？为什么？
2. 阅读程序，如果将两个函数中else if(*data==122) *data=’a’;和else if(*data==97) *data=’z’; 处的else 去掉，对程序有何影响？使用数据”I am a boy!”重新测试看看。
3. 仿造上例编写程序：设计一个带密钥的加密算法，例如密钥可以是一个常数，字符串加密的方法是将每个字符的ASCII码值加上该常数，然后对128求模。要求以密钥将加密的字符串加密输出，再以相同的密钥将加密字符串解密输出。

实验5.4　求数组中最大元素

求一个3×4矩阵中的最大元素，将求矩阵中的最大元素的过程定义为一个函数。函数的第一个参数是矩阵本身，第二个参数是第一维的大小。这种方法的优点是使函数具有通用性，即无论一个矩阵的第一维是多大，只要该矩阵的第二维是4 个元素，都可用该函数求最大元素；也可用该函数求一个矩阵开始几行中的最大元素。

【程序】

代码清单 5.3　*

```
#include <iostream>
using namespace std;
int max_value(int array[][4], int n);

int main(){
    int a[3][4]={{1,3,6,7},{2,4,6,8},{15,17,34,12}};
    cout<<max_value(a,3)<<'\n';
    return 0;
}
int max_value(int array[][4], int n){
    int i, j, max = array[0][0];
    for(i=0;i<n;i++)
        for(j=0;j<4;j++)
            if(array[i][j]>max)
                max=array[i][j];
    return (max);
}
```

【要求】

1. 将max_value(int array[][4], int n)改为max_value(const int array[][4], int n)，程序还能运行吗？这样做有什么用？
2. 修改上述程序使其不仅可以求矩阵中的最大元素，还能求最大元素的行数和列数。

实验5.5　引用起返回值作用

编写一个函数，其原型为：void Index(int a[], int n,int & sub)，其功能是，在大小为n 的数组a 中，查找某个数sub，若找到，将其下标存放在sub 中，若没找到，将-1 存放在sub中，在主调函数中通过判断值来判断数组中是否有该数。在这里，sub 是引用类型的参数，但起返回值的作用。

【程序】

代码清单 5.4　*

```
#include <iostream>
#include <stdlib.h>
using namespace std;
void Index(int array1[], int size, int & sub);
int main() {
   int array2[25] = { 2,3,5,7,11,13,17,19,23,29,31,37,\
                      41,43,47,53,59,61,67,71,73,79,83,89,97 };
   int size = 25, sub;
   cin >> sub;
   Index(array2, size, sub);
   if (sub != -1)
      cout << "对应元素下标为:" << sub << endl;
   else
      cout << "未找到。" << endl;
   system("pause");
   return 0;
}
void Index(int array[], int size, int & sub) {
   int temp = sub;
   sub = -1;
   for (int i = 0; i<size; i++)
      if (array[i] == temp) {
        sub = i;
        break;
      }
}
```

【要求】

1. 修改主程序连续查找数字的循环语句，使程序能在输入特定数字后退出循环。
2. 将void Index(int array[], int size, int & sub); 改为void Index(int array[], int size, int sub);程序还能正确执行吗？试分析其结果并解释。
3. 在上述修改的基础上如果结果不正确，怎样修改可以同样得到正确的结果。

实验5.6　返回值引用

一个声明为返回引用的函数，既可以作为右值出现在赋值号的右边，也可以作为左值出现在赋值号的左边。下面是一个函数调用本身作为左值的例子。

代码清单 5.5　*

```
#include<iostream>
#include <stdlib.h>
using namespace std;
int& index(int array[], int num) {//定义索引函数
    return array[num];
}
int main() {
    int Array[] = { 2,4,6,8,10 };
    index(Array,3) = 16;
    for (int i = 0; i<5; i++)
        cout << index(Array,i) << ' ';
    cout << endl;
    system("pause");
    return 0;
}
```

【要求】

修改程序使用非返回引用的函数实现上述功能，试分析两者之间的区别。

实验5.7　函数的重载

重载函数允许不同的函数使用相同的名字，这使得完成类似的任务时可以使用相同的函数名。编写几个计算面积的函数，分别计算圆、矩形、梯形和三角形的面积，计算边长为1的正方形及其内切圆、内接等腰三角形和等腰梯形面积。

【提示】

1. 圆面积，参数为半径：double area(double radius=0)，默认参数为0，表示点面积；
2. 矩形面积，参数为长和宽：double area(double a, double b)；
3. 梯形面积，参数为两底和高：double area(double a, double b, double h)；
4. 三角形面积，参数为三边长，double area(double a, double b, double c, int)，int 型参数起标识作用，以区别于梯形，不参加计算。

【程序】

代码清单 5.6　*

```
#include<iostream>
#include<cmath>
using namespace std;
const double PI = 3.14159;
double area(double radius = 0);
double area(double length, double height);
double area(double bottom, double top, double height);
double area(double edge1, double edge2, double edge3, int);
int main() {
    cout << "点的面积为 " << area() << '\n';
    cout << "矩形的面积为 " << area(1,1) << '\n';
    cout << "圆的面积为 " << area(0.5) << '\n';
    cout << "梯形的面积为 " << area(1,0.5,1) << '\n';
    cout << "三角形的面积为 " << area(1,sqrt(1+0.5*0.5),
    sqrt(1 + 0.5*0.5),0) << '\n';
    return 0;
}
double area(double radius) {//带默认参数的函数
    return PI*radius*radius;
```

```
20 }
21 double area(double length, double height) {
22     return length*height;
23 }
24 double area(double bottom, double top, double height) {
25     return (0.5*(bottom+top)*height);
26 }
27 double area(double edge1, double edge2, double edge3, int) {
28     double s = 0.5*(edge1+edge2+edge3);
29     return sqrt(s*(s-edge1)*(s-edge2)*(s-edge3));
30 }
```

【要求】

1. 编译运行程序，并记录运行结果，注意函数调用时，实参与形参之间的关系（包括类型、个数）。
2. 若将计算矩形面积的函数原型改为double area(double length=0, double height=0); 重新编译运行情况会怎样？为什么？
3. 若将计算三角形面积的函数原型改为double area(double edge1, double edge2，double edge3); 程序还能正确运行吗？为什么？
4. 若将计算三角形面积的函数原型改为double area(double edge1, double edge2，double edge3=0，int); 程序还能正确运行吗？为什么？
5. 将本实验以多文件方式进行组织，在area.h 中声明各个area()函数原型，在area.cpp中定义函数，然后在Exp9_2.cpp中包含area.h,并定义main()函数并执行。

实验5.8　用递归函数实现勒让德多项式

勒让德多项式的分段函数形式如下：

$$P_n(x)=\begin{cases}1 & \text{n=0}\\ x & \text{n=1}\\ \dfrac{(2n-1)P_{n-1}(x)-(n-1)P_{n-2}(x)}{n} & \text{n>1}\end{cases} \tag{5.1}$$

【要求】

1. 利用递归函数求出勒让德多项式；
2. 在主函数中输入$P_4(1.5)$，求值。

第6章　类

实验目的

类是C++扩展数据类型，可以封装不同类型的数据成员和函数成员。类是面向对象程序设计的基础。本章实验内容包括面向对象的基本概念、构造函数与析构函数，从实际问题抽象出类等，通过实验要求掌握以下内容：

1. 掌握面向对象的基本概念和类的定义方法。
2. 掌握类成员的访问权限以及访问类成员的方法。
3. 掌握内联函数和默认函数。
4. 掌握构造函数和析构函数的意义及使用方法。
5. 学会编写与应用复制构造函数。
6. 掌握运算符重载成为友元函数的方法。
7. 掌握运算符重载成为成员函数的方法。

实验内容

实验6.1　定义一个矩形类

设计并测试一个矩形类（Rectangle）。属性为矩形的左上角与右下角的坐标，矩形水平放置。操作为计算矩形的周长和面积。

【提示】

矩形所在坐标系参考屏幕：左上角为坐标原点，向右为x轴正向，向下为y轴正向。

【程序】

代码清单 6.1 *

```
1. 头文件rect.h
#ifndef RECT_H
#define RECT_H
#include<iostream>
using namespace std;
class Rectangle {
    int m_left, m_top;
    int m_right, m_bottom;
public:
    Rectangle(int left = 0, int top = 0, int right = 0,
    int bottom = 0);
    ~Rectangle(){} //析构函数，在此函数体为空
    void show();//显示左上角与右下角坐标
    void assign(int left, int top, int right, int bottom);
    int area();//计算矩形面积
    int perimeter();//计算矩形周长
};
#endif

2. 源文件rect.cpp
#include "rect.h"
Rectangle::Rectangle(int left, int top, int right, int bottom) :
   m_left(left), m_right(right), m_top(top), m_bottom(bottom) {}
   //构造函数，带缺省参数，缺省值全为0，在声明中指定

void Rectangle::show() {//显示左上点和右下点的坐标
   cout << "left-top point is (" << m_left
      << "," << m_top << ")" << '\n';
   cout << "right-bottom point is (" << m_right
      << "," << m_bottom << ")" << '\n';
}
void Rectangle::assign(int left, int top, int right, int bottom) {
   m_left = left;
```

```
34      m_right = right;
35      m_top = top;
36      m_bottom = bottom;
37  }
38  int Rectangle::area() {
39      return (m_right - m_left)*(m_bottom - m_top);
40  }
41  int Rectangle::perimeter() {
42      return 2*((m_right - m_left)+(m_bottom - m_top));
43  }
44
45  3. 源文件main.cpp
46  #include "rect.h"
47  #include <stdlib.h>
48  int main() {
49      Rectangle rect;
50      rect.show();
51          rect.assign(100, 200, 300, 400);
52          rect.show();
53      Rectangle rect1(0, 0, 200, 200);
54      rect1.show();
55          rect1.assign(100, 200, 300, 400);
56      cout << "Area of rect:" << rect.area()
57         << " Perimeter of rect:" << rect.perimeter() << endl;
58      system("pause");
59      return 0;
60  }
```

【要求】

1. 将Rectangle(double left=0, double top=0, double right=0, double bottom=0)改为Rectangle(double left, double top, double right, double bottom)，程序仍能正确运行吗？为什么？
2. 注意成员函数show、area、perimeter的使用。因为在前面如果需编写类似功能的一般函数是需要带参数（形参）的，而在此处作为类的成员函数又不需要带参数。思考为什么？
3. 理解void assign(double left,double top,double right,double bottom)函数的作用。

将Rectangle(double left=0,double top=0,double right=0,double bottom=0) 改为Rectangle(double left,double top,double right,double bottom)，这时，有人认为Rectangle(double left,double top, double right,double bottom)和void assign(double left,double top,double right,double bottom) 的功能相同，那么assign 函数能否去掉呢？请试一试，结果会怎样？

4. 为Rectangle类添加复制构造函数Rectangle(const Rectangle & rhs)。

实验6.2 定义复数类

复数形如z=a+bi，a为实部，b为虚部。支持一般的四则运算，但不支持大小比较。【要求】

1. 定义复数类的默认构造函数，复制构造函数，析构函数；
2. 重载输出运算符“<<";
3. 设置实部和虚部。

【程序】

代码清单 6.2 *

```
#include<iostream>
using namespace std;
class Complex {
public:
    Complex()=default;
    Complex(double real,double imag);
    double real() { return m_real; }
    double imag() { return m_imag; }
    void set(double, double);
    friend ostream& operator<<(ostream &os,const Complex &rhs);
private:
    double m_real;
    double m_imag;
};
```

实验6.3　重载运算符

为上面实验6.2定义的Complex类重载运算符，并实现Complex类对象的计数功能int number()。

【要求】

1. 重载+，-，*，/，=，==，!=运算符，请注意哪些运算符可以作为类的成员函数，哪些需要作为类的友元；
2. 定义具体的复数类对象，实现运算符的操作测试；
3. 计数器用于检测到目前为止Complex类对象的个数，如

```
Complex a;
cout<<a.number()<<endl; //输出1
Complex b;
cout<<Complex::number()<<endl; //输出2
```

实验6.4　定义一个集合类

集合是具有同一属性（共性）而又能互相区别（个性）的多个成员汇集起来的整体，构成集合的每个成员称为集合的元素，元素间没有顺序关系。例如，所有的小写英文字母是一个集合，它包括26个元素：a、b、…、z。不包含任何元素的集合称为空集合。自定义一个集合类Set，采用vector存放集合的元素。自定义集合类有以下功能。Set类头文件和主函数测试代码已给出，请自己完成Set头文件中函数的定义。

【要求】

1. 判断元素elem是否为集合set的元素。
2. 为集合添加一个元素elem。
3. 从集合中删除一个元素elem。
4. 赋值和复制构造函数。
5. 显示集合中的所有元素。
6. 求两个集合中相同的元素，即求两个集合的交集。
7. 求两个集合中所有的元素，即求两个集合的并集。

【程序】

代码清单 6.3 *

```
1. 头文件Set.h
#include<iostream>
#include<vector>
using namespace std;
class Set {
   vector<char> m_elems;  //数据成员
public:
    Set() =default;
    Set(const vector<char> &elem);//构造函数
    bool is_elem(char);     //是否为集合元素
    void insert(char);  //插入一个元素
    void erase(char);       //删除一个元素
    Set common(const Set &s);    //两个集合的交集
    Set sum(const Set &s);       //两个集合的并集
    Set& operator =(const Set &s); //赋值运算符
    Set(const Set &s); //复制构造函数
    friend ostream& operator<<(ostream &os,const Set &s);
};
2. 源文件main.cpp
#include "set.h"
int main() {
    vector<char> temp1 = {'a','s','d','f','g'};
    vector<char> temp2 = { 'a','c','v','f','t','y','e','r' };
    Set s1(temp1), s2(temp2),s3,s4;
    s1.is_elem('a');
    s1.insert('p');
    cout << "s1={"<<s1<<"}"<<endl;
    s2.erase('t');
    cout << "s2={" << s2 <<"}" << endl;
    cout<<s2<<endl;
    s3=s1.common(s2);
    s4=s1.sum(s2);
    cout << "s3={"<<s3 <<"}"<< endl;
```

```
34      cout << "s4={" << s4 << "}"<<endl;
35      Set s5(s1);
36      cout << "s5={" << s5 << "}"<<endl;
37      s5 = s4;
38      cout << "s5={" << s5 << "}"<<endl;
39      return 0;
40  }
```

课程设计上　学生成绩管理系统

课程设计内容

C++语言，面向对象的分析与设计。

课程设计基本要求

学生成绩管理是高等学校教务管理的重要组成部分，主要包括学生成绩的录入、删除、查找及修改、成绩的统计分析等等。请设计一个系统实现对学生成绩的管理。
系统要求实现以下功能：

1. 增加记录：要求可以连续增加多条记录。
2. 删除一个学生的记录：要求可以先查找，再删除。删除前，要求用户确认。
3. 成绩修改：若输入错误可进行修改；要求可以先查找，再修改。
4. 查找：可以根据姓名（或学号）查找某个学生的课程成绩，查找某门课程成绩处于指定分数段内的学生名单等等。
5. 统计分析：对某个班级学生的单科成绩进行统计，求出平均成绩；求平均成绩要求实现函数的重载，既能求单科的平均成绩，又能求三科总分的平均成绩。求出一门课程标准差和合格率。
6. 排序功能：要求按总分进行排序（从高到低），若总分相同，则按数学排序；若总分和数学相同，则按物理排序；若总分和各科成绩都相同，则按学号排序。
7. 文件操作：可以打开文件，显示班级的所有学生信息；可以将增加或修改后的成绩重新写入文件；可以将排序好的信息写入新的文件。

较高要求

查找可以实现模糊查询，即输入名字的一部分，可以列出满足条件的所有记录。再从这个

记录中进行二次选择。

测试数据

一个文本文件（学生成绩数据.TXT）

实现提示

1. 在系统管理模块，不能用STL对教师和学生记录进行操作，其他地方可以使用STL。
2. 所有操作都要通过可视化界面来操作和显示

实现提示

设计一个学生的类和一个管理学生成绩的类。学生类的设计请参考测试数据的记录和将要实现的功能。学生一条记录包括：学号、姓名、数学成绩、物理成绩和英语成绩。学生成绩管理的类应能管理所有学生的成绩，包括上面要求的7种基本功能，可用vector 来存储所有学生信息。主函数显示功能菜单，供用户选择操作。每步操作之前，都要显示菜单。在主函数中调用类的方法。
学生类和成绩管理类的数据成员如下：

代码清单 6.4　*

```
class Student{
private:
    string m_id,m_name;
    int m_math,m_eng,m_phy;
};
class Management{
private:
    vector<Student> stu; //存储所有学生的信息
};
```

第7章　模板与泛型编程

实验目的

所谓泛型编程，就是以独立于任何特定类型的方式编写代码。这使得程序员在编写充分可重用代码时有了趁手的工具。泛型其实是一种形式的静态多态，实现方式是类型参数化。一旦类型本身被参数化，那么我们就可以跃过类型限制带来的鸿沟，用一个类或者函数操纵多种类型不同的对象，并且不需要知道实现的细节。这无疑为代码的编写和维护带来的巨大的好处。

1. 掌握泛型编程，并能基本运用模板的语法。
2. 培养理论联系实际和自主学习的能力，提高程序设计水平。

实验内容

实验7.1　用模板实现两个对象值的交换

设计一个函数模板，用来实现两个对象值的交换。并用string和double类型进行测试。【程序】

代码清单 7.1　测试代码

```
1 template<typename T>
2 T& Swap(T&a,T&b) {//系统也提供swap函数，所以自定义函数名首字母大写
3     T c = a;
4     a = b;
```

```
    b = c;
    cout << a <<" "<< b << endl;
    return a;
}

int main() {
    double i, j;
    string s1, s2;
    cout << "Please input two values:" << endl;
    cin >> i >> j;
    Swap(i, j);
    cout << "Please input two strings:" << endl;
    cin>>s1 >> s2;
    Swap(s1, s2);
    return 0;
}
```

实验7.2　将集合类改造为集合类模板

在实验6.4已经定义了一个集合类，在这里将其改造为集合类模板，并添加新的功能如下。

【要求】

1. 增加查找功能：find()，返回所找元素在集合中的地址；
2. 清除所有成员clear()；
3. 返回集合中元素个数；
4. 与另一个集合类实现内容交换swap()；
5. 补充Set类模板成员函数的定义，并用string，int，complex类型测试

实验7.3　设计MyVector类模板

仿照标准模板库中vector模板，利用数组（长度1000）设计一个MyVector类模板，要求实现以下功能(注意const和&的使用)。

【要求】

1. 默认构造函数，带参数构造函数，拷贝构造函数，赋值运算符。
2. size(), operator[](), at(), front(), back()。
3. push_back(), pop_back(), insert(), erase(), clear()。
4. sort():用冒泡排序法对集合元素进行从小到大排序；
5. 实现二分查找算法binary_search。

第8章　动态存储内存与数据结构

实验目的

1. 理解运行时内存分配的概念，掌握自由存储区内存动态分配的方法。

2. 理解内部包含为指针动态分配内存的类对象复制时的浅复制和深复制的概念，会编写深复制构造函数和赋值复制运算符。

实验内容

实验8.1　再设计MyVector类模板

实验7.3通过定长数组设计了一个MyVector类，但是其不具有vector类的可变容量属性。在这里，利用动态数组重新设计一个MyVector类，考察动态内存的使用。

【要求】

1. 默认构造函数，带参数构造函数，拷贝构造函数，赋值运算符，移动构造函数，移动赋值运算符；

2. 完成size(), operator[](), at(), front(), back(),push_back(), pop_back(), insert(), erase(), clear()等成员函数的重新定义；

3. 注意动态内存的使用与分配。

【提示】

利用动态数组设计的vector类模板如下：

代码清单 8.1　*

```
template<typename T>
class MyVector{
    T* m_arr;
public:
    MyVector(){};
    MyVector(const T *arr=nullptr);//默认构造函数
    MyVector(const MyVector &rhs);    //复制构造函数
    MyVector(MyVector &&rhs);         //移动构造函数
    ~MyVector(){delete [] m_arr;};    //析构函数
    T & operator[](int i);           //重载取下标运算符
    int size()                       //输出元素个数
    MyVector & operator=(const MyVector &rhs);//赋值运算符
    MyVector & operator=(MyVector &&rhs);//移动赋值运算符
    friend ostream& operator<<(ostream &os,const MyVector &rhs);
    ...
};
```

请读者完成各函数的定义，并测试。

实验8.2　基于链栈实现简单计算器

链栈是一种采用链式结点（node）结构实现栈（stack）后进先出（LIFO），只能在顶部结点进行操作的数据结构。基于链栈结构实现一个简单计算器的功能。

【要求】

1. 支持加、减、乘、除、求余、括号等基本操作；
2. （选做）：扩展计算器功能，使其支持sin、cos、tan、sqrt、pow等功能。
3. 基于设计的计算器，计算以下表达式：
   3-2*4+(6-1)/2+5，sin(30)，cos(45)，tan(60)，sqrt(9)，pow(2,3) 用于测试。

【提示】

1. 基本四则运算书本已经介绍；

2. 函数功能的实现可以考虑将运算符设为string类型压入运算符栈；
3. 函数实现关键是如何准确读取string类型的函数名，如“sin”，而不是“si”或“s”，这样才能确保得到所要的函数。

【程序】

代码清单 8.2 *

```
...
if (m_opr.top() == "sin")
    m_num.push(sin(c));
else if(m_opr.top() == "cos")
    m_num.push(cos(c));
else if(m_opr.top() == "tan")
    m_num.push(tan(c));
else if(m_opr.top() == "sqrt")
    m_num.push(sqrt(c));
else if(m_opr.top() == "pow")
    m_num.push(pow(c,d));
    ...
```

第9章　继承与多态

实验目的

继承与派生是面向对象的特性，是面向对象程序设计模拟客观世界的手段之一，本实验的内容包括介绍基础类的派生关系，通过实验要求掌握以下内容。

1. 掌握类继承与派生关系以及实现方法，理解类的层次结构。

2. 掌握派生类构造函数初始化基类成员和对象成员的方法。

3. 掌握赋值兼容原则，掌握派生类的复制构造函数和赋值运算符的定义。

4. 在掌握继承与派生关系的基础上，进一步理解虚函数与多态性的关系，实现运行时的多态。

5. 学会定义和使用纯虚函数。

实验内容

实验9.1　定义一个继承与派生关系的类体系

定义一个继承与派生关系的类体系，在派生类中访问基类成员。先定义一个点类，数据成员为x，y坐标；以点为基类派生一个圆类，增加表示半径的数据成员；采用组合与包含的方式定义一个线段类，以两个点类对象作数据成员。

【要求】

1. 主要考察派生类的拷贝成员的控制，包括构造与析构，复制、赋值、移动等操作；

2. 建立工程，录入上述程序，改变数据实验之；

3. 修改Point类的数据成员m_x,m_y的访问权限为private，再运行，结果如何？

4. 如果不将Segment类设为Point类的友元，应采取什么措施？为哪个类增加数据或函数成员？

【程序】

代码清单 9.1 *

```
1、头文件shape.h
#include<iostream>
using namespace std;
class Segment;
class Point {
   friend class Segment;
protected:
   double m_x,m_y;
public:
   Point() { cout << "default constructor" << endl; };
   Point(double x, double y) :m_x(x), m_y(y) {
      cout << "parameter constructor" << endl;
   }
   Point(const Point &p) :m_x(p.m_x), m_y(p.m_y){//复制构造函数
      cout << "copy constructor" << endl;
   }
   ~Point() { cout << "destruct point" << endl; }
   Point& operator=(const Point &p) {                    //重载=
      if (this != &p) {
         m_x = p.m_x;
         m_y = p.m_y;
      }
      cout << "point endow" << endl;
      return *this;
   }
};
class Circle :public Point {
protected:
```

```
29      double m_radius;
30  public:
31      Circle() { cout << "default construct circle" << endl; }
32      Circle(const Point &p, double r) :Point(p), m_radius(r) {
33          cout << "parameter construct circle" << endl;
34      }
35      Circle(const Circle &c) :Point(c), m_radius(c.m_radius){
36          cout << "copy construct circle" << endl;
37      }                                       //派生类复制构造函数
38      ~Circle() { cout << "destruct circle" << endl; }
39      Circle& operator=(const Circle &c) {//重载=
40          if (this != &c) {
41              m_x = c.m_x;
42              m_y = c.m_y;
43              m_radius = c.m_radius;
44          }
45          cout << "circle endow" << endl;
46          return *this;
47      }
48  };
49  class Segment {
50  protected:
51      Point p1;
52      Point p2;
53  public:
54      Segment() { cout << "default construct segment" << endl;}
55      Segment(const Point&a, const Point &b) :p1(a), p2(b) {
56          cout << "parameter construct segment" << endl;
57      }
58      Segment(const Segment &s) :p1(s.p1), p2(s.p2) {//复制构造函数
59          cout << "copy construct segment" << endl;
60      }
61      ~Segment() { cout << "destruct segment" << endl; }//析构函数
62      Segment& operator=(Segment &s) {                //重载=
63          if (this != &s) {
64              p1 = s.p1;
```

```
65          p2 = s.p2;
66      }
67      cout << "segment endow" << endl;
68      return *this;
69   }
70 };
71 2、源文件
72 #include "shape.h"
73 int main() {
74    {
75     Point p1(2, 3), p2(3, 4), p3(p1), p4;
76    Circle c1(p1, 3), c2(c1), c3;
77    Segment s1(p1, p2), s2;
78    s2 = s1;
79    p4 = p2;
80    c3 = c1;
81     }
82    return 0;
83 }
```

实验9.2　理解虚函数的作用

了解“单界面，多方法”的概念。现有称为figure的基类，存放了各二维对象（三角形、矩形两个类）的各维数据，set_dim()设置数据，是标准成员函数。area()为虚函数，因为计算各对象的面积的方法是不同的。

【程序】

代码清单 9.2　*

```
1 #include <iostream>
2 using namespace std;
3 class figure{
4 protected:
5     double m_x,m_y;
6 public:
7     void set_dim(double i,double j=0){
```

```
        m_x=i;
        m_y=j;
    }
    virtual void area(){//定义area()为虚函数
    cout<<"本类没有面积计算。\n";}
};
class triangle :public figure{
public:
    void area() //派生类中重写area()函数
    {cout<<"高和底为"<<m_x<<"和"<<m_y<<"的三角形面积为:"<<0.5*m_x*m_y<<endl;}
};
class square :public figure{
public:
    void area() //派生类中重写area()函数
    {cout<<"长和宽分别为"<<m_x<<"和"<<m_y<<"的矩形的面积为:"<<m_x*m_y<<endl;}
};

int main(){
    figure *p;    //定义一个基类指针
    triangle tri;
    square squ;
    p=&tri;       //基类指针指向派生类对象
    p->set_dim(10.0,5.0);
    p->area();
    p=&squ;
    p->set_dim(10.0,5.0);
    p->area();
    return 0;
}
```

【要求】

1. 建立工程，录入上述程序，调试运行并记录运行结果。
2. 修改上述程序，将virtual void area()中的virtual去掉，重新调试运行观察结果有何变化？为什么？
3. 修改上述程序入口函数，使其动态建立三角形、矩形2个对象，通过基类指针访问这2个

对象，然后释放这2个对象。

实验9.3　继承与组合

类之间存在着继承与组合的关系，利用继承与组合的完成：点、线、圆、三角形、矩形、圆柱、长方体、圆锥类的定义。

【提示】

1. 定义Shape基类和Point基类，其他形状类通过继承和组合得到；
2. 定容量容器类可以用一个vector实现，其元素为基类指针，基类指针指向派生类。

【要求】

1. 圆柱体、长方体、圆锥具有价值属性。
2. 以上类具有面积area，体积volume成员函数，且将重载operator<用于比较大小。（线：默认为长度大小；面：面积大小；体：体积比较）。
3. 设计一个具有一定容量的容器类：
   (a) 能容纳圆柱体、圆锥、长方体；
   (b) 按照体积排序；
   (c) 在给定物品当中选择一部分物品放入容器，使选中的物品总量不超过容器容量的前提下价值总和最大（不考虑物品的形状）。测试数据如下，最优方案中1代表对应物品被选择，0代表对应物品没有被选择。

测试数据			
容器容量	物品的容量	物品价值	最优方案
104	25,35,45,5,25,3,2,2	350,400,450,20,70,8,5,5	1,0,1,1,1,0,1,1

第10章　标准输入输出

实验目的

1. 了解常用IO类的继承关系和理解IO流基本工作流程。
2. 掌握常见的输入输出格式控制。
3. 掌握文件流和字符流的使用方法。

实验内容

实验10.1　简单输入输出

解背包问题，从以下文件中读取数据，并把计算结果写入文件result.txt中（测试数据放在data目录下面）。

【分析】

背包问题：给定一组物品，每种物品都有自己的重量和价格，在限定的总重量内，我们如何选择，才能使得物品的总价格最高。下面表格中为三组测试数据的文本文件。

【测试数据】

测试数据				
	容器容量	物品的容量	物品价值	最优方案
dataset1	p01_c.txt	p01_w.txt	p01_p.txt	p01_s.txt
dataset2	p02_c.txt	p02_w.txt	p02_p.txt	p02_s.txt
dataset3	p05_c.txt	p05_w.txt	p05_p.txt	p05_s.txt

第11章　标准模板库

实验目的

1. 理解迭代器的工作原理和使用方法。
2. 理解常见容器的特点并掌握使用它们的方法。
3. 了解算法的类型并掌握常用调用对象的使用方法。

实验内容

实验11.1　电话薄管理

手机是我们便捷的通讯工具，手机应用里最常见的一个功能就是电话薄，用来管理用户的电话号码。最常见的操作有：新建联系人，删除联系人，搜索联系人，添加号码，删除号码等。

【要求】

1. 利用multimap容器设计一个简单的用户，关键字为姓名（称呼）、值为号码；
2. 新建联系人时，关键字和值至少有一个输入；
3. 搜索联系人时，显示其所有号码；若无此联系人，显示为空；
4. 删除联系人时，可以选择删除某个号码，也可以选择将联系人信息全部清除；

课程设计下　学生选课和课程管理系统

课程设计内容

C++语言，面向对象的分析与设计及可视化实现。

课程设计基本要求

学生选课和成绩管理是高等学校教务管理的重要组成部分，主要包括教师管理学生成绩模块、学生选课模块和系统管理模块。每门课程包括学分、学时、课程名字以及课程性质（必修和选修）等信息。每位教师可以教授若干门课程，并负责学生成绩的录入、删除、查找及修改、成绩的统计分析等等。教师的信息还包括姓名和ID。学生根据自己的专业要求进行选课，比如需要完成的总学分(学校要求总学分为50)和必修课等情况。学生的信息还包括姓名、ID和班号等。请设计一个系统实现对学生的选课和成绩的管理。系统要求实现以下功能：

1. 教师模块：
   (a) 成绩的录入，要求从文件读取。
   (b) 成绩修改：若输入错误可进行修改；要求可以先查找，再修改。
   (c) 查找：可以根据姓名（或学号）查找某个学生的课程成绩，查找某门课程成绩处于指定分数段内的学生名单等等。
   (d) 统计分析：对某个班级学生或所有选课的学生的单科成绩进行统计，求出平均成绩，标准差和及格率；
   (e) 排序功能：对某个班级学生或所有选课的学生的单科成绩由高到低进行排序
2. 学生模块：
   (a) 根据当前学分和课程性质选择相应课程
   (b) 退选某些课程的学习
   (c) 查看所选修课程的成绩和当前选修总学分

3. 系统管理模块：
   (a) 学生入学或引进新教师时增加学生或老师的功能
   (b) 学生毕业或老师离职时删除学生或老师功能
   (c) 增加或删除某一门课程信息
   (d) 学生、教师或课程信息发生变动后，将结果保存到相应的新建文本里面。

附加功能

增加学生对教师的评价模块：学生对所选修课程进行评语并给出满意度成绩；教师模块可以查看学生评语和评分；系统管理模块根据学生的平均满意度对教师教学效果进行排序

测试数据

学生信息(student.txt)、学生成绩记录（score.txt）、教师信息(staff.txt)和课程信息(module.txt)，所有文件均以#END为结束标识

实现要求

1. 在系统管理模块，不能用STL对教师和学生记录进行操作，其他地方可以使用STL。
2. 所有操作都要通过可视化界面来操作和显示

实现提示

1. 可用自己实现的类std::vector类模板来对教师和学生记录进行动态管理；
2. 请考虑教师、学生和课程之间的关系，建立类结构关系；

附录　课程设计报告模板

课程名称
课程设计报告

班级：

学号：

姓名：

原创性声明：

本人声明报告者中的内容和程序为本人独立完成，引用他人的文献、数据、图件、资料均已明确标注出。除标注内容外，不包含任何形式的他人成果，无侵权和抄袭行为，并愿意承担由此而产生后果。

作者签字：　　时间：

课程成绩：

程序界面	程序结构和功能	程序安全和效率	报告	总分
10%	40%	40%	10%	

指导教师签字：　　时间：

报告正文部分

一　课程设计题目与要求

包括题目与系统功能要求。

二　需求分析

包括问题描述、系统环境、运行要求等。

三　概要设计

包括系统流程设计、系统模块设计等。

四　详细设计

包括类的函数成员和数据成员设计、界面设计（见问题提示）、及其它模块设计与实现。

五　测试

包括对各功能模块的测试。

六　结论

对系统开发的总结，存在的不足，需要改进的地方。

七　附录

程序源代码，关键代码要做注释。

报告打印要求

A4双面打印，正文用五号字体，但涉及到程序代码的地方请用小五打印。