

面向对象及 C++ 程序设计习题

一、顺序以及分支程序

- E01.** 编写程序计算用户输入的 3 个整数的平均值、和、最大值和最小值，并输出到屏幕上。输出格式要求平均值占 8 列，保留 4 位小数，左对齐；最大值和最小值占 6 列，右对齐。
- E02.** 从键盘输入一个字母，如果输入的是小写字母，请将其转换为大写字母输出，如果输入的是大写字母，请将其转换为小写字母输出，其余情况原样输出，请编写程序实现。
- E03.** 已知 2017 年 1 月 1 日是星期日，输入 2017 年的一个月和日，输出该天是星期几。
- E04.** 一只大象口渴了，要喝 20 升水才能解渴，但现在只有一个深 h 厘米，底面半径为 r 厘米的小圆桶(h 和 r 都是整数)。问大象至少要喝多少桶水才会解渴。(注：小圆桶的深 h 和底面半径 r 请用户输入，单位默认是厘米。)
- E05.** 在苏大校园里,没有自行车,上课办事会很不方便.但实际上,并非去办任何事情都是骑车快,因为骑车总要找车、开锁、停车、锁车等,这要耽误一些时间。假设找到自行车、开锁并骑上自行车的时间为 27 秒；停车锁车的时间为 23 秒；步行每秒行走 1.2 米,骑车每秒行走 3.0 米。请判断走不同的距离去办事,是骑车快还是走路快。程序的输入是一个整数表示距离，单位是米，如果计算结果是步行快，请输出“步行快”，如果计算结果是骑车快，请输出“骑车快”，否则输出“一样快”。

二、循环程序

- E06.** 编写程序打印出 ASCII 码从 32 - 126 的 ASCII 字符（每行打印 5 个）。
- E07.** 求 $S_n = a + aa + \dots + aa\dots a$ 的值。其中 a 是一个数字。 a 和 n 都是由键盘输入。例如：
- 求 $S = 2 + 22 + 222 + 2222 + 22222 + 222222$ ，那么 $a = 2$ 且 $n = 6$ 。
- E08.** 程序员小明年薪 N 万，他希望在天赐庄买一套 55 平米的房子，现在价格是 200 万，假设房子价格以每年百分之 k 增长，并且该程序员未来年薪不变，且不吃不喝，不用交税，每年所得 N 万全都积攒起来，问第几年能够买下这套房子？（ N 和 k 请用户输入， N ($10 \leq N \leq 50$)， K ($1 \leq K \leq 20$)），如果在第 20 年之前（含 20 年）就能买下这套房子，则输出一个整数 M ，表示最早需要在第 M 年能买下，否则输出 Impossible。
- E09.** 公元前 5 世纪，我国数学家张丘建在《算经》中提出百钱百鸡问题：鸡翁一值钱五，鸡母一值钱三，鸡雏三值钱一，百钱买百鸡，问鸡翁、母、雏各几何？
- E10.** 校长将粉笔作为工资，发放给认真的教师。第一天，教师收到一支粉笔；之后两天（第二天和第三天）里，每天收到两支粉笔；之后三天（第四、五、六天）里，每天收到三支粉笔；之后四天（第七、八、九、十天）里，每天收到四支粉笔……这种工资发放模式会一直这样延续下去：当连续 N 天每天收到 N 支粉笔后，教师会在之后的连续 $N+1$ 天里，每天收到 $N+1$ 支粉笔（ N 为任意正整数）。

请编写一个程序，确定从第一天开始的给定 n 天数内 ($n < 500$)，教师一共获得了多少粉笔。

三、数据类型和数组

E11. 用随机数初始化一个 10 个元素的数组，每个元素的值都是 2 位数，请输出该数组元素的最大值、最小值和平均值。

E12. 从键盘输入一个不包括空格的仅由英文字母组成的字符串，长度不超过 200 个字符。统计 26 个英文字母的使用频率（出现的次数），不区分大小写。最后按使用频率从大到小的顺序输出字母（小写字母）及其使用频率，没有出现的字母不输出。

E13. 用手机发短信，一条短信资费为 0.1 元，但限定一条短信的内容在 70 个字以内（包括 70 个字）。如果你一次所发送的短信超过了 70 个字，则会按照每 70 个字一条短信的限制把它分割成多条短信发送。假设如下数组存储了一个人一个月发送的每个短信的分别字数

```
int array [12]={35,146,57,13,224,35,99,68,113,79,88,46},
```

试统计一下他当月短信的总资费。

E14. 已知数组 `int array [10]={13,24,35,46,57,35,79,35,93}`，请将用户输入的一个数字按照从小到大的顺序插在对应的位置后输出整个数组。然后将数组中重复的元素去除后输出。

E15. 让用户输入 n 个整数，存储到一个向量中，然后对该向量的值按逆序重新存放。例如，原来的顺序为 8,6,5,4,1。要求改为 1,4,5,6,8。 n 和整数的值请用户输入。

E16. 编写程序对 12 个月的英文单词根据字典序进行从小到大的排序，请使用 string 向量实现。

E17. 手工在程序所在目录下创建 1 个文本文件 Data.txt，该文件中存放了 9 个 int 数据，各数据之间以空格间隔。编写 1 个程序，使用文件流的相关方法顺序完成以下操作：

- 读取文件 Data.txt 中的所有数据；
- 将读取得到的数据以二进制的形式存放到程序所在目录下的文件 Res.dat 中；
- 再次读取 Res.dat 中的所有数据；

● 并将这些数据以文本文件的形式存放到程序所在目录下的 Res.txt 文件中，存放时每行 2 个数据，每个数据占 10 列，右对齐，左边补'C'字符。

四、函数

E18. 写一个判断一个整数是否是素数的函数 IsPrime。在主函数中求 1000 以内的所有质数，每行 8 个输出到屏幕上。函数原型要求为：

```
int IsPrime(int Num);
```

E19. 编写一个函数 InverseBits，将一个整数的各位数字对调。在主函数中编写测试代码，输入整数和输出新的整数。例如：输入 123，调用该函数之后，得到结果为 321。函数原型要求为：

```
int InverseBits(int Num);
```

E20. 设计一个函数 Calc，完成如下功能。

- 接受两个整数作为参数。
- 计算出最大公约数和最小公倍数。(不要在 Calc 函数中使用输入输出流)
- 可以在函数参数中使用引用。
- 编写主函数，测试函数功能。
- 函数原型要求为：

```
void Calc(int Num1, int Num2, int &GCD, int &LCM);
```

E21. 编写三个重载函数 printOut 分别实现输出整型数、浮点型数和 string 字符串，输出要求如下：

- 整型数：占 10 列，右对齐
- 浮点型：有 4 位小数
- 字符串：占 10 列，右对齐
- 编写测试程序，测试程序中的整数、浮点数和字符串由用户输入。

五、类和对象

E22. **[第 8 章：类的基本操作]**设计一个类 Circle，表示圆形。

- Circle 类以圆心坐标(x,y)和半径 r 来确定圆。
- Circle 类具有如下方法：
 - i. 可设置圆心坐标。
 - ii. 可设置半径。
 - iii. 可计算圆的面积。

- iv. 可计算圆的周长。

编写主函数，测试类的功能。

E23. [第 8 章：类的基本操作]设计一个类 Column，表示圆柱体。设圆柱体底面在 $Z=0$ 的平面内。

- Column 类以底面圆（参见 E22 题）和高来确定圆柱体。
- Column 类具有如下方法：
 - i. 可设置底面圆心。
 - ii. 可设置底面半径。
 - iii. 可设置高。
 - iv. 可计算底面积。
 - v. 可计算底面的周长。
 - vi. 可计算侧面积。
 - vii. 可计算体积。
- 编写主函数，测试类的功能。

E24. [第 8 章：类的基本操作，运算符重载]在 E22 圆形类的基础上，完成如下功能：

- 定义加法运算，规则：两圆之和为一个新的圆，圆心是第一个操作数的圆心（如 $a+b$ ，则 a 的圆心为 $a+b$ 的圆心），半径为两圆半径之和。加法运算不改变操作数。
- 定义减法运算，规则：两圆之差为一个新的圆，圆心是第一个操作数的圆心，面积为两圆面积之差的绝对值。减法运算不改变操作数。

- 定义自增、自减运算（含前、后自增），对半径进行自增、自减运算。
- 定义输出流运算，输出圆心坐标、半径、周长、面积。
- 定义>、<运算，比较两圆面积之间的大小关系。
- 定义==、!=运算，比较两圆是否是完全相同的圆，当两个圆的圆心坐标和半径完全一致时则两圆完全相同。
- 定义&运算，确定两圆是否同心。
- 定义|运算，确定两圆的位置关系（相交、相切、相离、包含）。
- 编写主函数，任意生成若干圆，分别测试上述功能。
- **提示：可自行决定是否增加圆的属性。但是，不提倡为了简化函数的计算而无原则增加属性。这样做在给某些计算带来方便的同时，也可能使另外一些计算变复杂。如：在记录半径的情况下，再记录周长和面积，可以简化某些运算符重载函数的代码。但是，半径与周长、面积之间有确定的约束关系需要遵守。所以，增加了面积和周长属性后，改变其中任意一项的值都意味着需要重新计算另外两项的值，否则就会产生数据的不一致。**

E25. [第 8 章：运算符重载，友元函数]设有描述复数的类 CComplex 定义如下：

```
class CComplex
{
    double m_real;
    double m_image;
public:
    void setValue(double real,double image)
    {
        m_real=real;
        m_image=image;
    }
};
```

请使用友元函数实现如下重载：

- 重载<<运算符，使得可以用 `cout<<`输出复数，每个复数输出的格式为：
 - ◆ “实部+虚部*i”；
- 重载+运算符，使得可以实现两个复数相加；
- 重载+运算符，使得可以实现复数和实数的相加；
- 重载前置++运算符，使得可以实现复数的实部和虚部分别加 1；
- 重载后置++运算符，使得可以实现复数的实部和虚部分别加 1；
- 在 `main` 函数中测试并试用这些运算符。

E26. **[第 8 章：静态成员]**现需要处理银行活期存款业务，设账户类为，`CAount`，请根据如下需求实现该类，并在 `main` 函数中测试。

- 每个账户需要有一个浮点型的成员 `m_Money` 用于存储账上余额；
- 每个账户需要描述存款的日期；
- 银行的年利息采用浮点型静态数据成员 `m_InterestRate` 描述，从而避免为每个账户存储利息；
- 为年利息成员提供静态成员 `SetInterestRate` 进行设置；
- 为年利息成员提供静态成员 `GetInterestRate` 进行获取；
- 提供存款成员函数 `SaveMoney`；
- 提供取款成员函数 `LendMoney`；
- 提供计算利息函数 `CalcInterest`；
- 提供结算利息函数 `SaveInterest`，该函数将计算出的利息结算到本金中。

- 为简化计算，请考虑以下定义或限制：

(1). 本题目不考虑闰年，每个月都认为 30 天，一年认为 360 天。

(2). 存款仅考虑发生一次！

(3). 取款允许发生多次，但取款是否允许需要考虑“本金是否足够”的条件

- 银行相关业务和利息的计算方法举例如下：

(1). 假设年利率 $m_InterestRate=0.036$ （表示 3.6%）。 $m_InterestRate$ 是静态成员变量，按照静态成员变量的概念，对所有账户 `CAount` 类的对象而言 $m_InterestRate$ 只有 1 个，这样才能实现一改全改的效果！

(2). 我于 2014-1-1 到银行存了 100000 元， $m_Money=100000$ 。

(3). 2014-3-10 银行给我“结算利息”一次。2014-1-1 到 2014-3-10 之间一共间隔了 70 天，本金 $m_Money=100000+100000*0.036/360*70=100700$ 。“结算利息”以后，存款日期变为了 2014-3-10！

(4). 2014-3-30 我到银行想取款 200000，由于本金 m_Money 只有 100700，所以不允许取款！

(5). 2014-4-4 我到银行想取款 50000，由于本金有 100700，所以允许取款。2014-3-10 到 2014-4-4 之间一共间隔了 25 天，本次取得的金额是 $50000+50000*0.036/360*25=50125$ ，本金 $m_Money=100700-50000=50700$ （利息不从我的账户中扣除，这是由银行提供给我的回报），存款日期仍然维持为 2014-3-10

六、构造函数

E27. **[第 9 章：类成员，类成员的构造（冒号语法）]**下面的一段程序中的 `Cline` 类

中缺少了部分成员函数，该程序的运行结果如下：

```
Point 1 is:(0,0)
Point 2 is:(0,0)
Length=0
Point 1 is:(1,1)
Point 2 is:(5,5)
Length=5.65685
```

请为 Cline 函数补充必要的成员函数与实现代码，使得程序正确运行。

```
#include <iostream>
#include <cmath>
using namespace std;

class CPoint
{
    int m_x; //点的 X 坐标
    int m_y; //点的 Y 坐标 °
public:
    CPoint()
    {
        m_x=0;
        m_y=0;
    }
    CPoint (int x,int y)
    {
        m_x=x;
        m_y=y;
    }
    int getx()
    {
        return m_x;
    }
    int gety()
    {
        return m_y;
    }
    void showPoint()
    {
        cout<<"("<<this->m_x<<","<<this->m_y<<")"<<endl;
    }
};

class CLine
```

```

{
    CPoint m_point1;
    CPoint m_point2;
};

void main()
{
    CLine line1;
    line1.ShowLine();
    cout<<"Length="<<line1.distance()<<endl;

    CLine line2(1,1,5,5);
    line2.ShowLine();
    cout<<"Length="<<line2.distance()<<endl;
}

```

- E28. [第 9 章：类成员，类成员的构造（冒号语法）]**设有类 CTime 和 CDate 分别用于描述时间和日期，另外有 CDateTime 类描述日期和时间，请为三个类给出具体的实现代码，并在 main 函数中测试。

```

#include <iostream>
#include <cmath>
using namespace std;

class CTime
{
    int m_hour,m_mintue,m_second;
public:
    CTime(int hour=0,int minute=0,int second=0);
    //如果时间合法，则赋值，否则赋值 0 时 0 分 0 秒
    int SetTime(int hour=0,int minute=0,int second=0);
    //如果时间合法，则赋值并返回 1，否则不赋值，并返回 0
    int getHour();
    int getMinute();
    int getSecond();
    void ShowTime(bool flag);
    //flag 为 True 则以 24 小时制显示时间，否则以 AM PM 的方式显示
    //自己考虑是否需要添加其他成员函数
};

class CDate

```

```

{
    int m_year,m_month,m_day;
public:
    CDate(int year=2000,int month=1,int day=1);
    //如果日期合法，则赋值，否则赋值 2000 年 1 月 1 日
    int SetDate(int year=2000,int month=1,int day=1);
    //如果日期合法，则赋值并返回 1，否则不赋值，并返回 0
    int GetYear();
    int GetMonth();
    int GetDay();
    void ShowDate(bool flag);
    //flag 为 TRUE，则以中文的方式显示日期，否则以 MM/DD/YYYY 的方式显示日期
    //自己考虑是否需要添加其他成员函数
};

class CDateTime
{
    CTime m_time;
    CDate m_date;
public:
    //添加必要的构造函数

    //int SetDateTime(...); 自己设计参数，考虑该函数的返回值加以表示全部正确、日期不
    //对、时间不对等情况

    //void ShowDateTime(...); 自己设计参数，考虑可以中文日期和西文日期格式，以及 24
    //小时和 AM PM 都可以控制
    //添加自己认为必要的其他成员函数
};

```

在主函数中验证上述类的功能，要求的主函数如下所示，不允许进行修改：

```

void main()
{
    CDateTime d1(2014, 5, 2, 27, 12, 5);
    d1.ShowDateTime(false, true);    //第 1 个参数表示以英文方式显示日期，第 2 个参
    数表示以 24 小时制方式显示时间

    CDateTime d2;
    d2.ShowDateTime(true, false);    //第 1 个参数表示以中文方式显示日期，第 2 个参
    数表示以 AM PM 的方式显示时间

    int iStatus;
    iStatus=d2.SetDateTime(2014, 5, 2, 21, 55, 5);
}

```

```

switch(iStatus)
{
case 1:
    cout<<"日期和时间数据设置正确!"<<endl;
    break;
case -1:
    cout<<"日期数据设置不正确!"<<endl;
    break;
case -2:
    cout<<"时间数据设置不正确!"<<endl;
    break;
}
d2.ShowDateTime(false, false);    //第 1 个参数表示以英文方式显示日期, 第 2 个参
数表示以 AM PM 的方式显示时间

cout<<"月="<<d2.GetMonth()<<endl;    //认真考虑一下如何实现此操作?
cout<<"分钟="<<d2.GetMinute()<<endl;    //认真考虑一下如何实现此操作?
}

```

七、继承

E29. [第 10 章：继承，派生类的构造]有一个 person 类定义如下：

```

class CPerson
{
private:
    string  m_name;    // 姓名
    int     m_age;     // 年龄
    char    m_sex;     // 性别
public:
    CPerson(char *name, int age, char sex)
    {
        m_name = name;
        m_age = age;
        m_sex = sex;
    }
    CPerson()
    {
        m_name = "无名";
        m_age = 18;
        m_sex = 'M';
    }
    void print()

```

```

    {
        cout<<"\n 姓名: "<<m_name<<"\n 年龄: "<<m_age<<"\n";
        if (m_sex == 'M')
            cout<<"性别: 男"<<endl;
        else
            cout<<"性别: 女"<<endl;
    }
};

```

请以 CPerson 类为基类定义一个派生类 CStudent,要求该类具有以下属性成员和成员函数:

- 学生所属大学名称, string 类型表示
- 学生所在年级, int 类型表示
- CStudent(); // 以{"无名"、18 岁、男性、“苏州大学”、1 年级}为默认值的无参构造函数;
- CStudent(string name, int , char, string collageName, int grade); // 带参数的构造函数
- void print(); // 显示学生类对象的所有信息

编写对 CStudent 类的测试程序, 要求如下:

- (1) 定义一个学生类 student1, 属性值为默认属性值, 并输出其信息
- (2) 定义一个学生类 student2, 其属性为:

- 姓名: "Liming"
- 年龄: 21
- 性别: 男
- 大学: 苏州大学
- 年级: 1

- (3) 提供 CStudent 类的完整测试程序, 要求界面友好, 输出结果应该有相应的

提示信息。