求全班同学的平均加权成绩（友元函数）

成绩	10	开启时间	2024年03月10日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

定义一个学生类，有三个私有成员：名字name、课程学分grade、课程成绩score。定义一个友元函数，求全班同学的平均加权成绩。平均加权成绩=每门课的成绩*每门课的学分的乘积的和/所有课程学分的和；全班同学人数不超过100名。

输入说明：输入若干行。每行一个学生的信息：第一个输入是学生的名字，第二个输入是第一门课程的学分，第三个输入是第一门课程的成绩，第四个输入是第二门课程的学分，第五个输入是第二门课程的成绩，以此类推，最后以-1表示该行输入结束。每个学生的课程数不超过100门。最后以 no 表示输入结束。

输出一行，即该全班同学的平均加权成绩。

输出日期（友元函数）

成绩	10	开启时间	2024年03月10日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

设计一个日期类和时间类，编写display函数用于显示日期和时间。要求：display函数作为类外的普通函数，分别在Time和Date类中将display声明为友元函数。在主函数中调用display函数，display函数分别引用Time和Date两个类的对象的私有数据，输出年、月、日和时、分、秒。

四进制加法（运算符重载）

成绩	10	开启时间	2024年03月10日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

定义一个四进制的类，重定义“+”号实现四进制数的累加。

输入：第一行输入所需要的四进制数的个数
第二行开始，依次输入四进制数。

输出：所有输入四进制数累加的和。

货币加减运算（运算符重载）

成绩	10	开启时间	2024年03月10日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

定义一个RMB类 Money，包含元、角、分三个数据成员，友元函数重载运算符‘+’（加） 和 ‘-’（减），实现货币的加减运算。

日期输出（运算符重载）

成绩	10	开启时间	
折扣	0.6	折扣时间	
允许迟交	否	关闭时间	

重载运算法<<，使之能够使用cout将Date类对象的值以日期格式输出。

员工请假条

成绩	10	开启时间	2024年03月17日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

定义一个基类Person，它有3个protected的数据成员：姓名name(char 类型)、性别 sex(1表示男，0表示女)、年龄age(int 类型)；一个构造函数用于对数据成员初始化；有一个成员函数show()用于输出数据成员的信息。

创建Person类的公有派生类Employee，增加两个数据成员： 基本工资 basicSalary（int类型）和 请假天数leaveDays（int型）；为它定义初始化成员信息的构造函数，和显示数据成员信息的成员函数show()。

输入：依次输入姓名，性别（0或1），年龄，基本工资，请假天数。

输出：录入的信息。

家和办公室

成绩	10	开启时间	2024年03月17日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

一个建筑物（building）包含基本的信息，由该建筑物可以得到办公室（office）和家（house）；他们又有各自特殊属性。具体要求如下：
（1）定义一个基类building,内有数据成员：层数（floor：int 类型）、房间（room：int 类型）和面积（area：double 类型)。定义自己的构造函数和输出函数
（2）定义一个家类House，它是building类的派生类，在类中添加数据成员：Bedrooms（int 类型）、Bathrooms（int 类型），定义自己的构造函数和输出函数，在构造函数中继承基类的函数。
（3）定义一个办公室类Office，它是building类的派生类，在类中添加数据成员：tables（int 类型）,Phones（int 类型）
（4）定义主函数进行测试。
数据输入输出顺序格式如下：

输入：3 5 100 2 3

输出：house_information

floor:3

room:5

area:100

Bedrooms:2

Bathrooms:3

计算形状的面积（抽象类）

成绩	10	开启时间	2024年03月17日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

定义抽象基类Shape，由它派生出2个类: Circle (圆形)、Rectangle (矩形)，显示两个图形的面积（圆周率用3.14159代替）。

要求:

(1)抽象基类Shape的公有成员有纯虚函数area()。

(2) Circle类公有继承自Shape类，新增数据成员radius (半径)，公有成员有构造函数和求圆面积的area()函数。

(3) Rectangle 类公有继承自Shape类，新增数据成员length (长)、width (宽)，公有成员有构造函数和求矩形面积的area()函数。

(4)在main()函数定义Circle类的对象circle1并赋初值，调用area()函数显示该圆面积;定义Rectangle类的对象rectangle1并赋初值，调用area ()函数显示该矩形面积。

编写一个程序，其中有一个书类book，该类的数据成员包括：书号、书名、出版社和定价；有一个作者类author，该类的数据成员包括：姓名、年龄和写作时间，每个类都有相应的数据输入、输出。以此两个类为基类，派生出图书查询卡card，并增加一个数据成员表示书籍系统名称，及一个可以显示系统名称、书名、作者、作者年龄、出版社和定价等数据的函数。

注： 输入数据内容有
系统名称 图书编号 图书名 出版社 定价
作者姓名 作者年龄 写作时间
输入：
BitLibrary
1001
C++Language
Bitcon
24.8
Lichunbao
40
2001 10 10
输出：

SysName:BitLibrary
Num:1001
BookName:C++Language
BookConcern:Bitcon
Price:24.8
AuthorName:Lichunbao
AuthorAge:40
PrintTime:2001-10-10

人员信息录入（虚基类）

成绩	10	开启时间	2024年03月17日 星期日 12:00
折扣	0.6	折扣时间	2024年05月3日 星期五 12:00
允许迟交	否	关闭时间	2024年06月30日 星期日 12:00

定义一个人员类People，其属性（保护类型）有：姓名、性别、年龄；

从People类派生出学生类Student，添加属性：学号和入学成绩；

从People类再派生出教师类Teacher，添加属性：职务、部门；

从Student和Teacher类共同派生出在职研究生类GradOnWork，添加属性：研究方向、导师；

分别定义以上类的构造函数、输出函数及其他函数（如需要）。

输入：

第一行：姓名，性别，年龄；

第二行：学号，成绩；

第三行：职务，部门；

第四行：研究方向，导师；

输出：各类的成员输出，具体可见实例。

定义一个抽象类Shape，在此基础上派生出正方体类、球体类和圆柱体类，都有计算对象表面积和体积的函数Area()和计算对象体积的函数Volume ()，在主函数中定义一个Shape指针数组分别指向正方体类、球体类和圆柱体类的对象，并通过Shape类的指针实现对这3个对象的成员函数的调用，输出正方体、球体和圆柱体的表面积和体积。

输入：第一行：正方体边长；

第二行：球体半径；

第三行：圆柱体的半径，高度。

输出：

Shape：正方体

面积、体积：** **

Shape：球体

面积、体积：** **

Shape：圆柱体

面积、体积：** **

定义猫科动物Felid类，由其派生出猫类（Cat）和豹类（Leopard）。

Felid包含构造函数、析构函数和纯虚函数sound

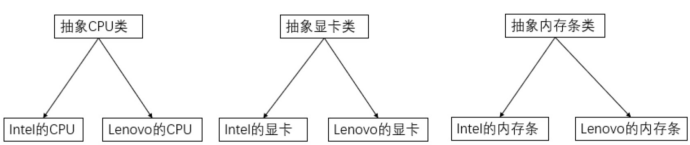
Cat包含构造函数、析构函数和虚函数sound

Leopard包含构造函数、析构函数和虚函数sound

要求：定义一个Felid类指针，采用动态内存分配的方式建议一个Cat对象，并且调用sound函数，之后撤销该对象，采用动态内存分配的方式建议一个Leopard对象，并且调用sound函数，之后撤销该对象。

提示：虚析构函数

计算机由cpu、内存条、存储器三种零件组成。生产零件的两大厂商 是intel和lenovo，每个厂商可以通过继承和多态来创建自己的cpu、内存条和存储器。继承关系建议可以用下图的方法设置：三个零件各自设置抽象基类，每个抽象基类都派生出了两个具体的子类，表示两种厂商的零件。计算机类的成员变量可以设置为三个基类的指针，指针根据要求可以指向不同厂商的零件。在主函数中，通过1和2选择厂商来设置计算机类中零件的指针，1代表intel,2代表lenovo。



输入输出示例如下：

输入：1 2 2

输出：

Intel的CPU开始计算了

Lenovo的显卡开始显示了

Lenovo的内存条开始存储了

某学校对教师工资的计算公式如下：固定工资+课时补贴。教授的固定工资为20000元，每个课时补贴50元；副教授的固定工资为15000元，每个课时补贴30元；讲师的固定工资为10000元，每个课时补贴20元。

要求：

定义教师抽象类，数据成员包括姓名，性别，工号，固定工资，课时数（上课总的时间），总工资；成员函数pay计算总工资，将其定义为虚函数（因计算方法不同），其他成员函数根据需要进行定义。派生出不同职称的教师类（每个类具有相同的数据成员）；不同类的工资计算方法不同（课时工资不同），编写程序求教师的工资。

编写一个程序实现小型公司的工资管理。该公司有4类人员，经理(manager)、技术人员(technician)、销售(sales)、销售经理(salesmanager)。这些人员都是职员(employee)，有姓名和月工资信息。月工资的计算方法是：经理固定月薪8000元，技术人员每小时100元，销售员按当月销售额的4%提成，销售经理既拿固定月工资5000元也拿销售提成，销售提成为所管辖部门当月销售额的5‰。要求编程计算月工资并显示全部信息。

经理Tom 技术人员John 销售经理Antony 销售Jane。

由键盘输入技术人员的工时数、销售经理的部门总销售额，销售员的销售额

编写一个模板函数outputArray，接收一个数组，将该数组所有元素倒序输出。部分代码已给出，请将代码填补完整。

```
int main() {

    int A_COUNT, B_COUNT, C_COUNT;
    cout << "请输入整型数组、浮点型数组、字符型数组的长度：" << endl;
    cin >> A_COUNT >> B_COUNT >> C_COUNT;

    int* arr_int = new int[A_COUNT];

    double* arr_double = new double[B_COUNT];

    char* arr_char = new char[C_COUNT];

    cout << "请输入长度为" << A_COUNT << "的整型数组：" << endl;
    for (int i = 0; i < A_COUNT; ++i)
        cin >> arr_int[i];

    cout << "请输入长度为" << B_COUNT << "的浮点型数组：" << endl;
    for (int i = 0; i < B_COUNT; ++i)
        cin >> arr_double[i];

    cout << "请输入长度为" << C_COUNT << "的字符型数组：" << endl;
    for (int i = 0; i < C_COUNT; ++i)
        cin >> arr_char[i];


    cout << "倒序输出整型数组：" << endl;

    outputArray(arr_int, A_COUNT);

    cout << "倒序输出浮点型数组：" << endl;

    outputArray(arr_double, B_COUNT);

    cout << "倒序输出字符型数组：" << endl;

    outputArray(arr_char, C_COUNT);
    return 0;
}
```

定义Store类，存放任意类型的数据成员item，并定义该成员的存入函数和提取函数。在主函数中，定义两个Store类的对象，其成员item分别为整型和浮点型，为每个对象的item成员赋值，并输出item成员的值。

预设代码

后置代码

view plain print ?

```
/* PRESET CODE BEGIN - NEVER TOUCH CODE BELOW */

int main() {

    int a;
    double b;
    cout << "请输入整变量a，以及浮点型变量b：" << endl;
    cin >> a >> b;

    Store<int> s1;
    Store<double> s2;
    s1.putElem(a);
    s2.putElem(b);
    cout << "s1.getElem() = " << s1.getElem() << "  " << "s2.getElem() = " << s2.getElem() << endl;

    return 0;
}

/* PRESET CODE END - NEVER TOUCH CODE ABOVE */
```

设计一个实现排序功能的函数模板，排序算法任意。要求排序结果由小到大，并在预设代码中验证int型数组和char型数组的排序结果

预设代码

后置代码

view plainprint?

```
/* PRESET CODE BEGIN - NEVER TOUCH CODE BELOW */

#include<iostream>
using namespace std;

int main()
{
    int data1[100];
        //= { 1,3,5,7,9,11,13,15,17,19,2,4,6,8,10,12,14,16,18,20 };
    char data2[100];
    int len_int;
    int len_char;
    int i = 0;
    int j = 0;

    cout << "请输入int型数组的长度" << endl;
    cin >> len_int;
    for (i = 0; i < len_int; i++)
        cin >> data1[i];

    cout << "请输入char类型数组的长度" << endl;
    cin >> len_char;
    for (j = 0; j < len_char; j++)
        cin >> data2[j];

    Sort(data1, len_int);//编写排序函数，data1是数组，len_int是数组长度
    Sort(data2, len_char);

    cout << "int型数组排序后：" << endl;
    for (i = 0; i < len_int; i++)
        cout << data1[i] << " ";
    cout << endl;

    cout << "char型数组排序后：" << endl;
    for (j = 0; j < len_char; j++)
        cout << data2[j] << " ";
    cout << endl;

    return 0;
}
```

编写一个模板函数getSum，接收两个不同类型的数组（int与double），返回各数组所有元素的和。

输入：每个测例共 3 行，第一行为两个整数 n（整形个数）,m（double型个数）(n > 1, m > 1)，第二行为 n 个整数，用空格隔开，第三行为 m 个实数，用空格隔开。

输出：对每个测例输出两行，第一行为输入的n个整数的和，第二行为输入的m个实数的和。

预设代码

后置代码

view plainprint?

```
/* PRESET CODE BEGIN - NEVER TOUCH CODE BELOW */

int main()
{
    int n, m;
    cin >> n >> m;
    int* arr_int = new int[n];
    double* arr_double = new double[m];
    for (int i = 0; i < n; ++i)
        cin >> arr_int[i];
    for (int i = 0; i < m; ++i)
        cin >> arr_double[i];
    cout << getSum(arr_int, n) << endl;
    cout << getSum(arr_double, m) << endl;
    return 0;
}
```

建立类模板input，在调用构造函数时，完成以下工作：

（1）分别给定数值的输出范围（通过设定最小值与最大值）

（2）输入一个待输出数字或字符，符合范围内(闭区间)便输出；不符合范围内显示 “数据不符合范围，请重新输入”。

输入格式：

第一行：数字的最小值，数字的最大值，待输出数字。

第二行：字符的最小值，字符的最大值，待输出字符。

输出格式：

第一行：待输出数字。

第二行：待输出字符。

预设代码

后置代码

view plainprint?

```
/* PRESET CODE BEGIN - NEVER TOUCH CODE BELOW */

int main() {
    int x, y, z;
    cin >> x >> y >> z;
    char a, b, c;
    cin >> a >> b >> c;
    input< int> in1(x, y);
    in1.output(z);

    input < char> in2(a, b);
    in2.output(c);

}
```

实现数组的循环左移，要求：构造ver 类，用于存储长度为 n 的整型数组，其中，print() 函数，用于输出数组中的元素； LeftMove() 函数，用于实现数组的循环左移操作。

输入格式:

第一行输入两个整数 n(1<n<100)和k(0≤k≤n)，分别表示顺序表的元素个数和循环左移的位移量。

第二行一共 n 个整数，表示顺序表中元素的值。

输出格式：

输出只有一行，输出 n 个整数，顺序输出循环左移后顺序表中每个元素的值，每个元素之间用一个空格分隔。行末不要有多余空格。

样例输入

8 3

1 2 3 4 5 6 7 8

样例输出

4 5 6 7 8 1 2 3

预设代码

后置代码

view plainprint?

```
/* PRESET CODE BEGIN - NEVER TOUCH CODE BELOW */

int main()
{
    int n, k;
    cin >> n;
    cin >> k;
    ver a(n);
    a.insert(n);
    a.LeftMove(n, k);
    a.print(n);
    return 0;
}
```

请实现一个链表：

insert x y：将y加入链表，插入在第一个值为x的结点之前。若链表中不存在值为x的结点，则插入在链表末尾。保证x,y为int型整数。

delete x：删除链表中第一个值为x的结点。若不存在值为x的结点，则不删除。

输入：

第一行输入一个整数n (1≤n≤104)，表示操作次数。

接下来的n行，每行一个字符串，表示一个操作。保证操作是题目描述中的一种。

输出：

输出一行，将链表中所有结点的值按顺序输出。若链表为空，输出"NULL"(不含引号)。

题目描述： 给定一个链表，判断该链表是否是回文结构。

例如： 1->2->3->2->1 是回文链表。

1->2->3->3->2->1 是回文链表。

输入：待判断的整数链表
输出：若是回文结构则输出1，否则输出0。

已知两个单链表 LA 和 LB 分别表示两个集合，其元素递增排序，设计算法求出 LA 和 LB 的交集 C ，要求 C 同样以元素递增的单链表形式存储。

用栈实现十进制转十六进制

假设一个算术表达式可以包含三种括号：“（”和“）”，方括号“[”和“]”，及花括号“ { ”和“ } ”，且这三种括号可嵌套使用。试设计算法判断给定表达式中所含括号是否配对出现。

给定一组入栈顺序和一组待定的出栈结果，要求判断出栈结果在给定的入栈顺序下是否可能。

输入格式：

第一行：输入整数n，表示栈的大小。

第二行：入栈顺序

第三行：待验证出栈结果。

输出格式：

判断是否可能。

编写一个程序，任意输入n个100以内的数，将它们的奇数和偶数分别存入链队为Q1和Q2中，然后配对输出链队Q1、Q2中的值，直到任一队列为空为止。

按先序遍历序列建立一个二叉树的二叉链表，并按先序遍历、中序遍历、后序遍历将其输出。其中#表示空树

按先序遍历序列建立一个二叉树的二叉链表，统计二叉树中叶子结点个数和二叉树的深度

设计算法，将n个数据组成二叉排序树结构，并可以删除其中的一个结点。
输入：数据个数n、n个数据、需要删除的数值value。
输出：原始数据、二叉排序树的中序输出及删除结点value后的结果。

先输入图的顶点数和边数，再输入图的各个顶点的编号，然后输入各条边的信息，最后输出图的邻接矩阵

已知一有向图，构建该图对应的邻接表。

邻接表包含数组和单链表两种数据结构，其中每个数组元素也是单链表的头结点，数组元素包含两个属性，属性一是顶点编号info，属性二是指针域next指向与它相连的顶点信息。

单链表的每个结点也包含两个属性，属性一是顶点在数组的位置下标，属性二是指针域next指向下一个结点。

输入：

第1行输入整数t，表示有t个图

第2行输入n和k，表示该图有n个顶点和k条弧。

第3行输入n个顶点。

第4行起输入k条弧的起点和终点，连续输入k行

以此类推输入下一个图

输出：

输出每个图的邻接表，每行输出格式：数组下标 顶点编号-连接顶点下标-…-^，数组下标从0开始。

具体格式请参考样例数据，每行最后加入“^”表示NULL。