选择题

执行以下语句： int x; char s[100]; cin>>x; cin.get(); cin>>s; 当用户输入：3（回车）A Sjtu（回车）后字符串 s 中的值为____C____

A、 “Sjtu” B、 “A Sjtu” C、 “A” D、 ‘A’

- 用cin>>输入后会在“输入缓冲区”留下一个回车符‘\n’，可以用cin.get()把这个回车符读走。
- cin>>读取字符串时把空格、回车等空白字符当做分隔符，每次读取都到分隔符的前一个字符为止，并在末尾加上‘\0’。如果缓冲区的第一个字符就是分隔符，会先把它删除。

选择题

cmath 库中一个求整数绝对值的函数原型为：int abs(int x)，试问以下哪种调用会报错____B____

A、double x=3.5; int y=abs(int(x));　　B、int x=3; int y=abs(int x);

C、int x=3; int y; y=abs(x);　　D、int x=3; int y=abs(x);

- 调用函数时，实参不能带数据类型。
- int(x)不是实参带类型，而是表示强制类型转换。也可以写作(int)x或(int)(x)，也就是说类型和变量至少一方要带小括号。

选择题

若有声明：char *s，之后运行下列语句，则哪个语句运行时不会出现错误____A____

A、 s =“SJTU”;　　B、s={‘S’, ‘J’, ‘T’, ‘U’, ‘/0’};

C、 s=strcpy(s, “SJTU”);　　D、s={‘S’, ‘J’, ‘T’, ‘U’};

- 选型A表示把一个**常量字符串的起始地址**赋值给一个字符指针。
- 选项B和D是给字符数组初始化的方式，不能用于赋值语句。（也不能用于字符指针的初始化语句。）
- 选项C的错误在于s是随机值，不能把字符串复制到随机地址上。

选择题

已知对指针 p 的定义，以下哪个声明可以使得 p 不得作为赋值语句的左值使用____B____

A、const int *p ;　　B、int * const p;　C、int const *p;　D、int *p const;

- 选项A和C表示p指向的对象不能被修改/赋值。
- 选项B表示p本身不能被修改/赋值。

选择题

任意一个类，关于构造函数和析构函数，哪个说法是正确的_____A_____

A、 构造函数可以有多个，析构函数只能有 1 个

B、 构造函数可以有多个，析构函数也能有多个

C、 构造函数只能有 1 个，析构函数也只能有 1 个

D、 构造函数只能有 1 个，析构函数可以有多个

- 构造函数可以有各种参数，可以根据参数来重载。
- 析构函数没有参数，无法重载，所以只能有1个。

选择题

整数在计算机内部中通常用_____C_____来表示。

A. 原码　　B. 反码　　C. 补码　　D. ASCII 码

- 原码：只能表示无符号整数。
- 反码：1换成0，0换成1。
- 补码：表示整数。
- ASCII码：表示字符。

选择题

已知变量 ch 的类型说明为 char ch; 则以下不符合 C++语法的表达式是____D____

A、ch='h'　　B、ch = 32　　C、ch = '\a'　　D、ch ='hi'

- 选项D明显错误，单引号内不能有两个字符。
- 选项C的‘\a’是响铃字符。

选择题

执行 int a = 1, b = 2, c = 2, d = 4; bool m=1, n=1; bool result=(m = a > b) && (n = c > d);
则 m, n, result 的值分别为__A____

A. 0, 1, 0　　B. 0, 0, 0　　C. 1, 2, 0　　D. 1, 0, 0

- 逻辑运算符 && 先执行左边的表达式，如果左边的表达式的结果为false，那么整个&&运算的结果就可确定为false，右边的表达式也就不执行了。
- 类似的，||左边的表达式如果结果为true，也就不再执行右边的表达式。

选择题

已知 int a[]={1,2,5,3} 下列哪一个表达式的结果为 3？ C

A. *(&a+2)　　B. *(a+2)　　C. *a+2　　D. *(&a[0]+2)

- 选项A的&a表示以整个数组a为一个元素的数组地址（即二维数组），+2表示下两行。
- 选项B和D是一样的。

选择题

当一个程序中调用库中的函数时，以下说法正确的是 __A__

A. 需 include 该库的头文件
B. 除了该库的头文件，不需要其它与该库相关的任何文件
C. 直接调用库函数就可以，程序中不需做任何与该库相关的操作
D. 一定要有库函数实现的源码文件

- 需要库的头文件和实现文件，但只include头文件。
- 实现文件不一定要是源码文件，可以是编译后的二进制文件。

选择题

已知 char str[10] = "2019\t"；则 strlen(str)+sizeof(str)的值是__B__ 。

A. 小于等于 14　　B. 15　　C. 16　　D. 17

- 字符串“2019\t”中有6个字符，4个数字+1个制表符+1个空字符。
- strlen函数从左往右数字符串中字符的个数，直到空字符为止（不包括空字符）。
- sizeof(数组名)返回该数组的整体大小，与其内容无关。

选择题

以下有关函数的说法，正确的是__ B ____ 。

A. 不能定义两个函数名一样的函数　　B. 两个函数的原型声明不能相同

C. 函数返回时所有局部变量都会自动被撤销

D. 参数传递是值传递，数组作为参数传递时，数组按元素逐一传递

- 不同类的成员函数可以同名、同参数，但它们的原型实际上是不同的，因为完整的原型还包含了 类名:: 。
- 数组作为参数传递时，只传递数组的起始地址，不传递任何元素。

选择题

若有如下定义，则对数组元素的成员引用不正确的是_C_____ 。

```
struct Student { int id;    char name[20];      char gender;        int age; };
Student stu [2], *p; p=stu;
```

A. p[1].gender　B. p->age　C. p[0]->id　D. stu[1].name

- 通过结构体指针或类指针访问成员用箭头（->），通过结构体变量或类对象访问成员用圆点（.）。
- 选项C的p[0]表示该结构体数组中的第0个元素，它是一个结构体变量，所以应该用圆点（.）来访问成员。

选择题

已知 A 类，则当程序执行到语句：A a[4],*pa[3];时，调用了____B____次构造函数。

A. 3　　B. 4　　C. 7　　D. 8

- 定义类的对象的时候才会调用构造函数。类的指针不是对象。
- A *p = new A;会调用构造函数，但这是new A的结果，与p无关。

选择题

设有整型数组 a[10]，要输出数组 a 的 10 个元素，错误的是：____B____

A. for (int *p=a, int i=0; i<10; ++i) cout << p[i];
B. for (int i=0; i<10; ++i) { cout << *a; ++a; }
C. for (int *p=a; p<a+10; ++p) cout << *p;
D. for (int i=0; i<10; ++i) cout << *(a+i);

- 选项B中的a是数组名，不能进行a=a+1的赋值操作，所以当然也不能执行++a。
- 选项C中的p是个指针，可以进行++p的操作，即p=p+1。

练习卷讲解

选择题

以下关于函数参数的叙述不正确的是：__B__

A. 函数的参数不仅仅可以是函数的输入，还可以是函数的输出
B. 对象作为实参时系统将自动调用拷贝构造函数
C. 函数的形参命名可以任意，只要符合标识符规则
D. 函数参数的值可以是内存单元地址

- 对象作为实参时，形参有两种情况：如果是值传递，则调用拷贝构造函数；如果是引用传递，则不创建新的对象，自然也就不调用拷贝构造函数。
- 对于选项A，如果函数的参数是引用传递，那么可以用于函数的输出（函数的输出指的是将函数的执行结果传递到函数外部，不一定要靠return）。
- 对于选项C，函数的形参名与普通变量名遵循相同的命名规则，即须符合标识符规则（只以字母、数字和下划线组成且不以数字开头）。

选择题

已知程序

```
#include <iostream>
using namespace std;
int main()
{ for(int i=5; i>1; --i) cout <<i; cout << i; return 0; }
```

则运行该程序之后，输出：____E____

A. 5432　　B. 54321　　C. 55443322　　D. 543210　　E. 以上都不对

- 该程序中的变量i是在for循环控制行中定义的变量，无法在循环结束之后访问。该程序中的第二个cout << i;在for循环之外，所以这个程序是无法编译的。

练习卷讲解

选择题

已知语句 myfun(x) = yourfun(y);可以正常运行，以下说法中正确的是：____C____

A. myfun 函数的参数只能采用引用传递
B. yourfun 函数的参数只能采用引用传递
C. myfun 函数不能返回非静态的局部变量
D. yourfun 函数不能返回非静态的局部变量

- 当函数调用是赋值表达式的左值时，它的返回类型一定是引用，而返回引用的函数不能返回非静态的局部变量，否则该变量会在函数返回后消亡，导致在主调函数中为该返回值取的“别名”失去意义，也无法被赋值。
- 对于选项D，当函数的调用语句是赋值表达式的右值时没有上述限制，返回类型可以是任意的。

练习卷讲解

选择题

以下关于异常的叙述，错误的是：__A__

A. 类的创建者必须能够发现和处理该类在使用过程中可能出现的异常

B. 如果 try 语句块中包含了可能抛出异常的代码，那么一旦抛出了异常，则程序退出 try 语句块，进入 try 后面的异常捕获和处理

C. 程序员可以将数组下标越界定义为一种异常

D. 使用 sin 函数但没有写#include<cmath>这种情况不属于异常

- 类的创建者只需要发现异常并将其抛出，而处理异常是类的使用者的职责。
- 至于选项D，这种情况是一种错误，但不是异常，它显然无法用异常处理机制来解决（库的创建者无法发现这种错误）。

选择题

以下程序段中，可以运行并且得到合理正确的结果的是：____D____

A. char s[]={'1','2','3'}; cout<<s;
B. int x=12345678; x*=x; cout << x;
C. int n; cin>>n; if(n=0) cout << "It is Zero"; else cout << "It is not zero";
D. 以上选项都不能得到合理正确的结果

- 在选项A中，s被定义为长度为3的字符数组，并未以空字符'\0'结尾，所以cout<<s可能输出跟在数组后面的非零数值，显示为乱码。

选择题

short int 类型的-1 在内存中存储为_____C_____

A. 0x0001　　B. 0xFFFE　　C. 0xFFFF　　D. 0x8001

- 0x0001是1，0xFFFE是-2，0xFFFF是-1。
- 对于有符号的整型数，“-N”的十六进制表示加上N必然等于0。

选择题

下面不合法的字符常量是_____D_____

A. '\\'　　B. '\012'　　C. '\0'　　D. "a"

- 题目本身有误导性，出题意图应该是“下面不是合法的字符常量的是”。
- ‘\\’是反斜杠的转义字符，‘\012’是第12号字符。
- 双引号表示**字符串常量**，单引号才表示**字符常量**。

选择题

下面的 do while 语句的循环体执行次数为____D____

```
unsigned int a = 10;
do {
    a--;
} while(a>=0);
```

A. 9　　B. 10　　C.11　　D. 无数次

- unsigned int表示自然数，所以 a>=0 恒成立。
- 当a为零0时，a--数据溢出，a的值变成2^32-1。

选择题

对于如下函数声明为，则下面调用写法正确的是____B____

```
void fun(int a, int b=1, char c='a', double d=3.2);
```

A. fun();　　B. fun(2,3);　　C. fun(2, ,'c', 3.14);　　D. fun(int a=1);

- 4个形参，后3个有默认值，因此1-4个实参均可。
- 但不能前面的实参留空，反而提供后面的实参。
- 函数调用语句中的参数为实参，实参不带数据类型。

选择题

B 是 double 常量

A. 01234　　B. 1e2　　C. 0xee　　D. 以上都不是

- 01234是八进制整型常量，0xee是十六进制整型常量。
- 1e2是用科学计数法表示的100。**科学计数法只能表示实型常量。**

选择题

以下说法正确的是____D____

A. C++提供的运算符按照运算数的个数分为两种：一元运算符和二元运算符
B. 逻辑运算中 1 表示为真，其它数值都表示假
C. 所有的运算符都可以重载，但是不能改变其运算优先级和运算数的个数
D. 不同类型的对象有些可以自动类型转换，例如可以将派生类对象赋给基类对象
E. 以上至少有两个叙述正确

- A错：还有个三元运算符 ?:（不能重载）。
- B错：只有0表示假，其它都表示真。
- C错：不是所有的运算符都可以重载。

选择题

若有函数 f(int*a, const int &b, int &c)以及定义 int m;和 const int n=9:，以下合法的函数调用是

A. f(&m, 5, 5)　　B. f(&m, 5, m)　　C. f(&n, n, n)　　D. f(m, m, n)

- A错在第3个参数：形参为非常量引用时，实参不能是常量。
- C错在第1个参数：形参为非常量指针时，实参不能是常量的地址。

练习卷讲解

选择题

以下关于函数的描述中说法正确的是___B____

A. 函数调用 f(1, , 'x')表示 f 的第二个参数将使用默认值

B. 处理逻辑完全相同而被处理的数据类型不同的多个函数可以写成一个函数模板

C. 重载函数是一组名字相同的函数，但是这些函数或者参数个数不同，或者参数个数相同而类型不同，或者参数完全相同但是返回类型不同

D. 以上至少有两个叙述正确

- A错：不能前面的实参留空，反而提供后面的实参。
- C错：仅有返回类型不同的函数是不能重载的。

选择题

要将一个二维数组 int a[3][4]传递给函数 f，以下函数原型____A____是正确的

A. void f(int[][4], int);　　B. void f(int[3][4]);

C. void f(int *, int);　　D. void f(int **, int);

- 以多维数组为实参时，形参的第一维留空，用另一个参数传递其数值。
- **二维数组的数组名是个一维数组指针**，不是普通的一级或二级指针。

练习卷讲解

选择题

以下哪些函数能够用于两个数的交换？ D

```
void swap1(int p, int q)
{
   int temp,
   temp=p;
   p=q:
   q=temp;
}
```

```
void swap2(int *p, int *g)
{
   int *temp;
   *temp=*p;
   *p=*q;
   *q=*temp;
}
```

```
void swap3(int *p, int *q)
{
   int *temp;
   temp=p;
   p=q;
   q=temp;
}
```

```
void swap4(int &p, int &q)
{
   int temp;
   temp=p;
   p=q;
   q=temp;
}
```

A. 只有 swap1　B. 只有 swap2　C. 只有 swap3　D. 只有 swap4

E. 有 2 个可以　F. 有 3 个可以　G. 都可以　H. 都不可以

- swap1参数是值传递。swap2的形参是p和g，交换的是p和q。
- swap3交换的是两个形参指针的已值，不是他值，因此不影响实参。

选择题

以下关于构造函数的 5 项描述中，不正确的个数是 D

1) 对于函数声明 void f(A &);当用 A 的对象 a 调用 f(a)时会自动调用拷贝构造函数
2) 拷贝构造函数的参数通常采用引用传递，但是也能够使用值传递，只是效率低一些
3) 如果有类 A，则语句 A a[4], *p; 会调用构造函数 4 次
4) 在对象之间互相赋值时会自动调用拷贝构造函数
5) 如果程序员未创建，系统可以自动生成一个无参数的构造函数和一个拷贝构造函数

A. 0　　B. 1　　C. 2　　D. 3　　E. 4　　F. 5

- 1错：引用传参不涉及构造。2错：拷贝构造函数的参数不能值传递。
- 4错：赋值用的是赋值运算符，不是拷贝构造函数。

运行题

```
#include <iostream>
using namespace std;
int main()
{     int m,n=0;
      for(m=1;m<=5;m++)
      {    switch(m)   {
           case 1:
           case 3:
           case 5:
           case 7: n=30; break;
           case 2: n=28;
           default: n=31; break;
           }
           cout<<n<<" ";
      }
      return 0;
}
```

- switch程序块中的语句是从匹配的case开始向下执行，直到遇到break为止。

运行题

```
#include <iostream>
using namespace std;
int   main()
{
    char a [20] = "C++_Programming";
    char *p = a;
    int i = 0;
    while (*p) {
        cout << "*p[" << i << "]=" << *p << endl;
        p = p + 3;    i++;
        }
    return 0;
}
```

- 每个周期中p往后移3个元素，直到*p的值为零（即p指向空字符为止）。
- 每个周期中i的值加1，但cout输出的不是p[i]而是*p，别看错了。

运行题

```
#include <iostream>
using namespace std;
void fun(int x, int *y, int &z)
{    x=*y+z;
     *y=x+z;
     z=x+*y;
}
int main()
{    int a=1, b=2, c=3;
     fun(a, &b, c);
     cout<<a<<" "<<b<<" "<<c<<endl;
}
```

- 注意参数的传递类型，其中x是值传递，所以形参的变化不影响实参。

练习卷讲解

运行题

```
#include <iostream>
using namespace std;
class Student {
        friend void fun (Student &rs);
        private:
                int num;
                double score;
                static int count;
        public:
                Student (int i=0, double j=0)
                        { num=i; score=j; count++;}
                void Print ()
                        {cout<< num<<','<<score<<endl;}
                static void disp(Student &ss) {cout<<"ss.count:"<<ss.count<< " count:"<<count<<endl;}
};
int Student::count = 0;
void fun (Student &rs) {
        rs.num=2018;
        rs.score=90;
        rs.Print ();
}
int main ()
{
        Student s1 (1008,60), s2;
        s1.Print ();
        fun (s2);
        Student::disp(s1);
        return 0;
}
```

- 因为count是静态成员，所以无论哪个对象的count都是一样的。

练习卷讲解

运行题

```
int main(){
    for (int i=0; i < 3; ++i){
        int j = 5;
        while(i < j){
            j = j - 3 - i;
            cout << '*';
        }
        switch (j) {
        case -1:
            cout << 'A';
            break;
        case 0:
            cout << 'B';
            break;
        case 1:
            cout << 'C';
        default:
            cout << 'E';
        }
    }
    return 0;
}
```

- i和j要看清楚。
- case -1和case 0后面都有break，而case 1后面没有break。

运行题

```
int *addone(int &arg){
        arg += 1;
        int *ptr = new int(arg);
        return ptr;
}
int main() {
        int a[5] = {1, 2, 3, 4}, *p, i;
        p = a;
        for(i=0; i<5; ++i) {
                p = addone(a[i]);
                a[i] *= *p;
                delete p;
                cout << a[i] << endl;
        }
        return 0;
}
```

- 函数addone的参数arg是引用传递，所以arg+=1会导致main函数中的实参a[i]的值加1。
- 函数addone返回是将局部变量ptr的值赋值给main函数中的p，这样p就会指向值为arg的动态整型变量。
- 当初始化表的长度小于数组长度时，剩余的数组元素被初始化为零。

练习卷讲解

运行题

```
class CC
{
private:
        int A,B;
public:
        CC(int i,int j)
        {A=i;B=j;cout<<"Call constructor\n";}
        CC(const CC &obj)
        {A=obj.A+1;B=obj.B+2;cout<<"Call copy constructor\n";}
        ~CC()
        {cout<<"Call destructor\n";}
        void print()
        {cout<<"A="<<A<<", B="<<B<<endl; }
};
int main()
{       CC a1(2,3);
        CC a2(a1);
        a2.print();
        CC *p = new CC(5,6);
        p->print();
        delete p;
        return 0;
}
```

- delete p会调用一次析构函数。
- main函数return时，main函数中的局部变量/对象都会消亡，所以a1和a2也要调用析构函数（析构的顺序与构造相反）。

运行题

```
int sum1(int a) {
    int d = 0;
    static int b = 5;
    d++;
    return (a + b + d);
}
int sum2(int a) {
    int c = 2;
    static int b = 15;
    b += 1;
    return (a + b + c);
}
int main() {
    int aa = 10;
    cout<<sum1(aa)<<','<<sum2(aa)<<endl;
    cout<<sum1(aa)<<','<<sum2(aa)<<endl;
    cout<<endl;
    return 0;
}
```

- 静态局部变量在第一次执行到其定义初始化语句时赋初值，在函数返回时其值保留不变，当再次进入该函数时跳过其定义初始化语句。

运行题

```
int main()
{
    int x = N;
    int y;
    switch (x/10) {
        case 0 : y = x; break ;
        case 1 : y = x/2+1.5; break ;
        case 2 : y = x/2+2.5;
        case 3 : y = x/2+3.5; break ;
        default : y = x/2;
    }
    cout<<y<<endl;
    return 0;
}
```

- x是整型变量，所以x/10和x/2都是整数运算，结果舍弃小数部分。
- y是整型变量，所以也舍去小数部分。
- case 2后面没有break。

运行题

```
int main()
{
	int a[]={2,1,0,-1,-2}, b[]={0,1,0,1,0}, c[5], i;
	for(i=0;i<5;i++) c[i]=a[i]&&(b[i]--);
	for(i=0;i<5;i++) cout<<a[i]<<' '<<b[i]<<' '<<c[i]<<endl;
	return 0;
}
```

- &&运算：左边为0时不执行右边。
- ||运算：左边为1时不执行右边。

运行题

```
char *myfun1 (char *p) { char *s = new char[strlen(p)+1]; return s; }
char &myfun2 (char a[ ]) { return a[0]; }
int main()
{
        char a[3][4] = {"one", "two", "six"}, *p = a[1];
        p = strcpy(myfun1(p), a[1]);
        cout << p << endl;
        char &b = myfun2(p);
        cout << &b+1;
        return 0;
}
```

- b是字符数组的第0个元素，则&b+1表示第1个元素的地址。
- 该地址是char*类型的，所以cout会输出从它开始的字符串。

运行题

常见考点：

- 整数运算（除法或赋值）的结果要舍去小数部分。
- switch语句中的case后面是否有break。
- 函数的局部变量或类的数据成员是否为static。
- 函数的参数为值传递，形参的变化不影响实参。
- 程序块结束时（如main函数return 0），局部对象要析构。

填空题

下面函数将一个整数字符串转换为一个整数。

```
int fun(char* str)
{
        int num, digital;

        num = 0;
        while(*str != 0){
                digital = *str-'0';
                num = num *10 + digital;
                str++;
        }
        return num;
}
```

- str[0]和*str是一样的。
- 0 和‘\0’是相等的。

练习卷讲解

填空题

```
#include <iostream>
using namespace std;
void getResult(_____int a[], int n, int &max, int &min, double *p_ave_____)
{
    _double total = 0;_____
    max = min = a[0];
    for (int i=0; i<n; i++)
    {
        if (a[i]>max) max = a[i];
        if (a[i]<min) min = a[i];
        total += a[i];
    }
    *p_ave =total/n;
}

int main()
{
    int a[5];
    int max, min;
    double ave;
    for (int i=0; i<5; i++)
        cin>>a[i];
    getResult(a, 5, max, min, __&ave__);
    cout<<ave<<" "<<max<<" "<<min<<endl;
    return 0;
}
```

- 要让被调函数中的修改影响主调函数中的变量，参数必须是指针传递或引用传递。
- 实参为变量，则形参为引用。
- 形参为指针，则实参为地址。

填空题

在不知道缺少什么代码的情况下：

- 如果是在析构函数或main函数结尾：是否需要释放动态内存？
- 如果是在类定义的开头：是否需要友元声明？
- 如果后面使用某个变量的值：
  - 该变量是否已定义/初始化/赋值？
  - 该变量是否通过参数传递进来？哪种传参方式？
- 如果后面用指针访问数组元素：指针是否已指向数组？
- 如果对类类型的对象使用运算符：运算符是否已重载？

编程题

输入一个字符串 str，再输入一个字符 c，试完成函数 void delChar (char str[], char c)的实现部分，该函数可将 str 中的字符 c 全部删除。

要求该函数用递归的方法来实现。

- 注意字符c是一个字符变量，不是字符‘c’。
- 将一个字符删除是指，去掉该字符后要把后面的部分往前移一格。
- 所以其实就是找到字符c的下标 i，然后 strcpy(str+i,str+i+1);
- 再递归调用 delChar(str+i, c); //注意不是str+i+1

练习卷讲解

编程题

定义一个日期类 TDate，使得下面测试程序

```
int main()
{
        TDate dt1(2,28,2020),dt2(12,31,2020);
        dt1.nextDay();
        dt1.print();
        cout << endl;
        dt2.nextDay();
        dt2.print();
        cout << endl;

        return 0;
}
```

运行结果为：

```
2/29/2020
1/1/2021
```

其中：函数 nextDay()实现求第二天日期的功能，函数 print()实现输出日期的功能。

- 闰年：((year % 4 == 0)&&(year % 100 != 0))||(year % 400 == 0)
- 别忘了每个月各有多少天，2月要分闰年和平年两种情况。
- nextDay除了要考虑下个月，还要考虑下一年。

编程题

设计一个支持整型、实型和字符型的冒泡排序的函数模板。调用方式参考以下代码片段：

```
int a[10] = {0,2,1,9,6,7,4,8,5,3};
char b[4] = {'a', 's', 'd', 'f'};
bubbleSort(a, 10);
bubbleSort(b, 4);
```

- 函数模板怎么写？最前面要写什么？自己拿张纸，不看书，试一试能不能写出来。
- 冒泡排序、选择排序、顺序查找、二分查找，都要复习一下代码。
- 在排序中，要清楚i和j的遍历范围，以及flag = false;、flag = true;和if(!flag)break;这三句在哪个大括号内、哪个大括号外。

练习卷讲解

编程题

设计一个程序，先让用户输入一个整数 num，然后让用户输入 num 行句子，每行句子的长度都小于 50。已知各行句子中只包含空格以及由字母组成的单词，句子的首个字符和末尾字符都是不是空格，相邻两个单词之间有且只有一个空格。要求该程序能将所有句子中最长的单词和最短的单词输出。如果有多个长度相同的最长单词或最短单词，输出最前面出现的那一个单词。部分程序代码如下，请写出函数 twoWords 的定义：

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

// 此处为函数 twoWords 的定义

int main() {
    int num;
    char *longest, *shortest;
    cout << "Please input the number of sentences: ";
    cin >> num; cin.get();
    cout << "Please input the sentences below: " << endl;
    twoWords(num, longest, shortest);
    cout << "The longest word is: " << longest << endl;
    cout << "The shortest word is: " << shortest << endl;
    //以下省略
}
```

- 函数twoWords有两个功能，一是把用户输入的num行字符串存入num个字符数组中，二是让longest和shortest这两个指针分别指向最长和最短单词的字符串（要以空字符结尾）。
- 功能二说明函数twoWords中要修改形参的值，所以必须是引用传递，即参数类型为char *&。
- 函数twoWords结束前要用delete释放不被main函数需要的堆内存空间。