期末考试说明

一、机考注意事项

1、请带好自己的有效证件（学生证/身份证），一卡通无效！

2、请提前至少 15 分钟到考场外候场！

3、考试时请严格遵守考场纪律，不要交头接耳、看其他同学的屏幕，否则因这些无谓的举动影响成绩，得不偿失！不要随身携带跟考试相关的资料、手机，一经发现均按作弊处理，成绩记零分！

4、填空题： 点击题板上的“答题”按钮进入 VS 环境，填空题中的**注释行“————space————”，不能删除！不要随便增加空行(系统在指定的行比对答案，如果增加空行，则找不到答案！！！)及增删程序其它位置的代码！**只需**删除填空的标志【？】填入语句即可！**注意标志后面是否有分号，如果有分号，只需删除该标志，在对应位置填入正确的语句即可（语句最后不加分号！）

例如， for(i=0; 【 ？ 】; i++) 中，填写的语句后面就不加分号。如果填空标志【 ？ 】后面没有分号，则填入的语句需要分号结束！

填空完成之后编译、改错、运行（如有样张，测试时可输入样张数据）。

5、改错题： 点击题板上的“答题”按钮进入 VS 环境，如果开

启错误提示，在题面中会看到有“————FOUND————”注释行，则错误就在该提示下方的一行语句中。先利用 VS 生成(编译)一遍，查看输出窗口的语法错误提示，将鼠标停留在输出窗口的 error 行，双击鼠标即可自动定位到程序的错误行！修改语法错误，之后再编译，无语法错误后运行。如果运行结果与样张不一致，说明还有逻辑错误，再进行修改，直到正确为止。（如有样张，测试时可输入样张数据）

6、编程题：点击题板上的“答题”按钮进入 VS 环境，在注释行“————————program————————”和“————————end————————”之间的空白处编码，不要删除这两个注释行，否则系统找不到你的答案！！！

作答时一定先认真阅读题板上的要求并严格按要求作答题目，了解题目中已有变量、数组、指针、函数等的作用。若给定相关的处理变量，则编程时只能用给定的变量，除非特别说明否则不能再自行定义新的变量；若规定了程序结构（如要求使用 for 循环，就不能用 while 或是 do-while 结构）则需要按要求编码。编码完成之后编译、改错、运行（如有样张，测试时可输入样张数据）。

7、特别强调：

- 填空、改错题，语句一定不要多加括号！
- 填空、改错题，一定不要随便增加空行及增删程序其它位置的代码！
- 表达式的写法一定要规范：变量一般写表达式左边，例如 i<N，或 i<=N-1。另外，涉及乘式的书写，要么常量系数全部在变量左边，要么全部在变量右边，　例如：100*i+10*j　或 i*100+j*10
- 函数声明时，形参名字可以省略(函数定义时形参名不能省略)，要么省略所有形参的名字，要么均不省略！

  int　max(int a, int b);　int max(int, int);

二、容易错误的地方

（1）常见的语法错误：

- 少头文件

  #include<string>　　//字符串类

  #include<cmath>　　//系统函数

  #include<cstdlib>

  #include<ctime>　　//随机数

  #include<iomanip>　//格式控制

- 判等 ==　　不等 !=　　大于等于　>=　　小于等于　<=
- 条件式中注意　与&& 和或|| 的区别

字符是否为元音字符：

(ch=='a'|| ch =='A'|| ch =='e'|| ch =='E'|| ch =='i'|| ch =='I'|| ch =='o'|| ch =='O'|| ch =='u'|| ch =='U')

字符是否为英文字母：

(ch>='a'&&ch<='z' || ch>='A'&&ch<='Z')

字符是否为数字：

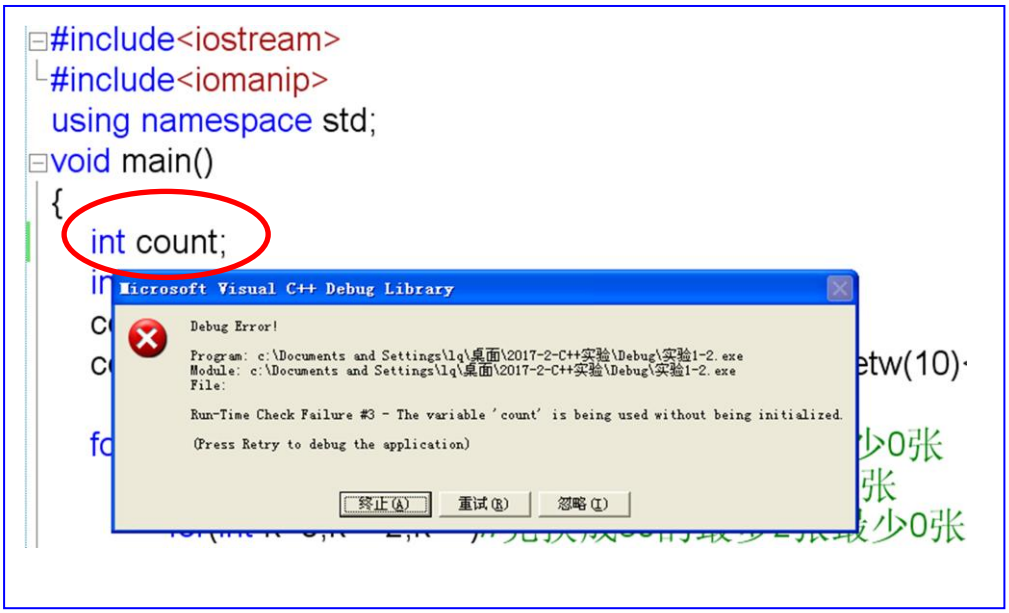
(ch>='0'&&ch<='9')

- break（退出最近的封闭的循环）和 continue（退出本次循环）的区别，如果改错题出现 break 和 continue 一定引起重视！
- int a=b=1;　(int a=1,b=1;)　40<a<50;　(a>40 && a<50)
- 注意 if /else 的配对关系　if 可以单独使用，但是 else 必须和 if 配对使用，else 总是和它上面的最近的未配对的 if 进行配对
- 数组元素下标一定从 0 开始，注意下标不要越界！元素在数组中的位置 pos 与元素的下标 i 差。
- 指针变量使用前一定初始化，其值只能是地址值！

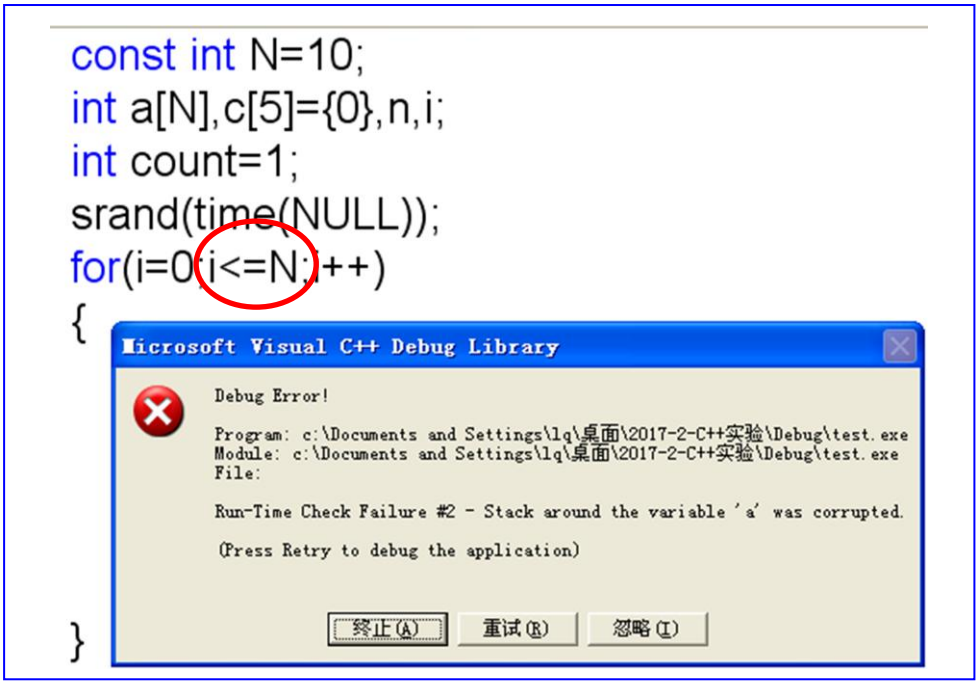
  for (p=a; p<a+N; p++)　cin>>*p;

  p=a;　//用 p 操作完数组后一定记着对 p 进行回溯！
- 求幂次的系统函数为 double pow(double, int);　不要写成 power。其参数一个为 double 型，一个为整型，如果数据类型不一致，可进行强制类型转换。如：　int x,y;　pow((double)x, y);
- 函数定义头部和函数声明语句完全一样！！！函数定义时，参数表的括号后面不能有分号！当数组作为函数参数时，数组名后面的方括号一定不能省掉！！！
- 函数调用时，实参和形成在类型和个数上要完全一致！如果是指针变量做函数的形参，如：　void fun(int *p);　则在调用时实参一定传递的是地址，可以是普通变量的地址，如 fun(&x); 可以是指向普通变量的指针，如 int *p; p=&x; fun(p); 或者是数组名 int a[N]; sort(a,N);或者是指向数组的指针，如 int *p, a[N]; p=a; sort(p,N);

（2）典型的逻辑错



看到这类运行时错误提示，仔细阅读提示信息，表示有未初始化的变量在使用，找到对应变量赋初值即可。



看到此类运行时错误提示，表示数组 a 被破坏了，即在访问 a 数组元素时下标越界，检查修改数组下标即可。

（3）编程题注意事项：

- **书中所有要求的案例代码一定熟练掌握!!!**
- 求平均数时，如果有小数点，一定将 sum 和 average 均设置为 double 型!

  其他问题求解时也请注意变量类型的选择!!!
- 输出格式：

  精度控制　cout.precision(10);　//整数部分和小数部分总位数，不包括小数点

  宽度设置 cout.width(n);　setw(n) ;//需引入头文件 iomanip

  小数点后位数

  cout<<fixed;

  cout.precision(n);　//两句一起使用
- 如果是产生随机数，请把随机数公式结果计算出来，例如 x=10+rand()%(100-10+1); 请写成：x=10+rand()%91;　//计算结果不加括号
- 编程时除非特别要求，一般不要使用逗号表达式!!!例如两个数的交换，t=x ; x=y; y=t; 不要写成 t=x, x=y, y=t;
- 不要乱加分号，编译时不会有语法错，也不是逻辑错，但是会影响运行结果!

  if(　); {…..}　　//相当于条件成立时执行一个空语句，该 if 语句与后面花括号中的语句是并列的关系

  while(　); {…..} //相当于循环条件成立时执行一个空语句，与后面花括号中的语句是并列的关系

  for(　); {…..} //相当于循环条件成立时执行一个空语句，与后面花括号中的语句是并列的关系
- 注意变量初始化及其放置位置!!!

  例如，求 n 个班中，各班 m 位同学的平均成绩

  for(i=0;i<n;i++)　　//n 个班

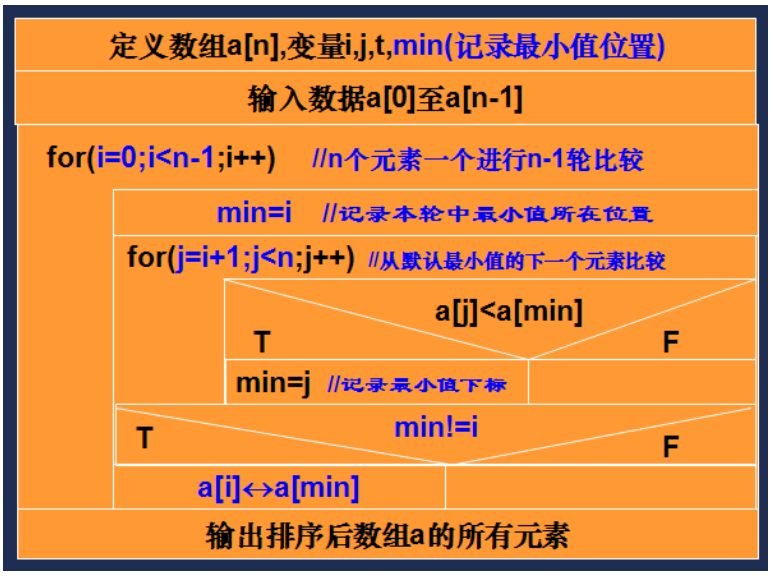
  {　sum=0;　//因为每求完一个班的总成绩，都需要初始化求和变量，注意其位置在两个嵌套的循环之间。更一般，二维数组处理时，若要对每行、

每列求和，求和变量的初始化都是放在两个嵌套哦的循环之间。

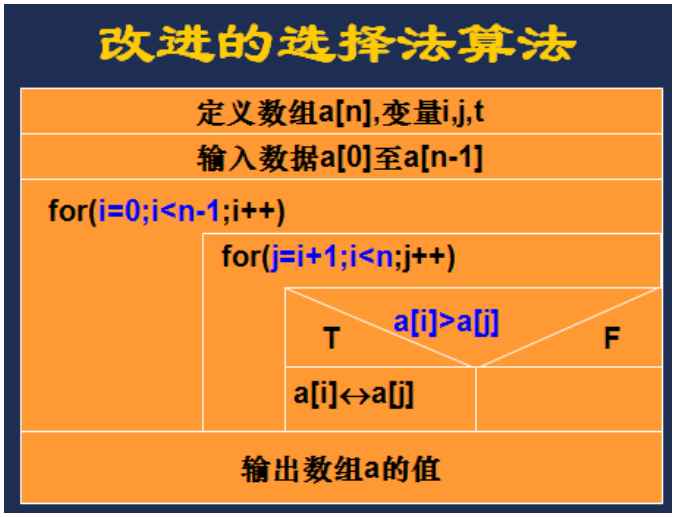
```
            for(j=0; j<m;j++)        sum=sum+a[i][j];
            average=sum/m;
    }
```

求一定范围内完数（sum=0 及其位置）、求一定范围内的素数（i=2 及其位置）等

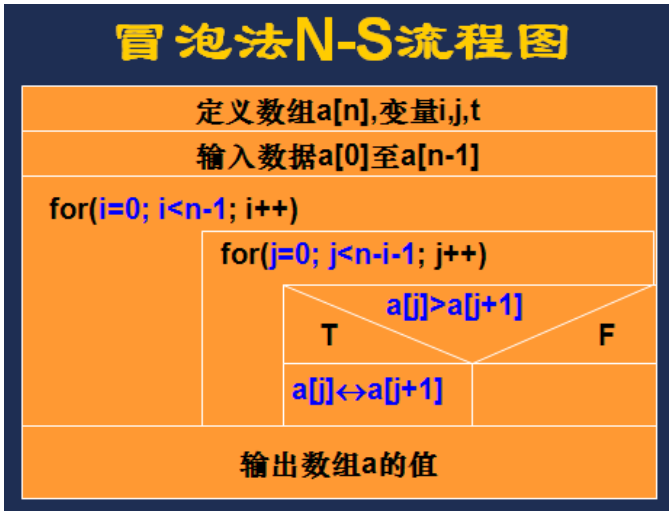
- 注意排序算法中，循环条件及比较条件的区别




```
for(i=0;i<n-1;i++)   //n 个数比较 n-1 轮
{
   min = i;   //初始化 min 的值
   for(j=i+1;j<n;j++)
     {
          if (a[j]<a[min])
               min=j; //记录最小值所在位置
     }
   if(min!=i)   //找到的最小值不是当前交换位置的元素才交换两数
     {
         t = a[i];  a[i] = a[min]; a[min] = t;          //交换 a[i]和 a[min]
     }          // 任意两数 a、b 实现交换（不用中间变量）：  a=a+b; b=a-b; a=a-b;
               //或写成:     a[i]=a[i]+a[min]; a[min]=a[i]-a[min]; a[i]=a[i]-a[min];
   }
```




```
for(i=0;i<n-1;i++)     //n 个元素比价 n-1 轮
    for(j=i+1; j<n;  j++)    //每轮处理里，从当前元素的下一个元素开始依次和当前元素做比较，如果比当前元素小则交换
        if(a[i]>a[j])
            { t=a[i]; a[i]=a[j]; a[j]=t; }          //排序
```




```
for(i=0; i<n-1; i++) //对 n 个数排序的轮数是 n-1
    for(j=0; j<n-i-1; j++)  //第 i 轮中,进行两两比较的次数是 n-i-1 次
        if(a[j]>a[j+1])
            {   t=a[j];     a[j]=a[j+1];     a[j+1]=t;   }
```

- 如果子函数的返回类型为 void，而又希望通过子函数中形参值的改变影响实参的值，则对应的形参可设置为引用变量（变量名前面加&，引用只能操作变量，不能操作数组）或是指针变量（变量名前面加*，指针变量既可操作变量，也可以操作数组）

用字符数组处理字符串

const int N=10;　char s[N];

- 输入时可以用 cin(不接收空格)，也可以用 cin.getline(s,N-1,'#')、cin.get(s,N-1), gets(s)（输出对应 puts）；
- 输出时，如果存储有结束符'\0'，则可整体输出 cout<<s<<endl; 否则用循环逐个输出字符元素或者在字符数组最后写入结束符

  s[N-1]='\0'; cout<<s<<endl;
- cin.getline(s,size)函数用于字符数组，getline 用于 string 类中，对字符串进行输入处理，改错时这两个函数容易换用
- 掌握字符数组对应的函数及其功能　　P128~P130
- l=strlen(s);　//strlen()用于字符数组，只取字符串的字符数，不取结束符

  for(i=0;i<l;i++)　　//元素下标从 0 开始
- 在字符串中找最大最小值时，默认最大和最小为 a[0]，和普通数组中的处理一样，最大最小值所在位置和元素下标的关系：两者之间相差 1。
- 奇数和偶数字符的统计，就是将输入的数字字符处理成普通数字，然后判断其奇偶性，'1'ASCII 码值为 49，故'1'-48=1，(a[i]-48)%2==0 为偶数，反之则为奇数。统计个数时一定初始化统计变量为 0，例如，sum=0；sum++
- 删除 ASCII 码值能被 m 整除的字符：凡是 ASCII 码值能被 m 整除(s[i]%m==0)的字符，则不处理(continue，跳过次字符的处理，不能用 break，否则结束整个处理)，否则输出。
- 对字符串中的字符进行排序也可以用选择和冒泡排序算法！掌握两种排序算法的核心部分。

用 string 类处理字符串

#include<string>　　//不能少

string s;

- 可以用 cin 输入（不接收空格）；也可以用 getline(cin,s)，接收空格；getline(cin,s,'#')，以特定字符作为输入结束的标志。getline(cin,s)，以回车换行作为输入结束的标志
- l=s.length();　l=s.size();　　for(i=0;i<l;i++)
- 删除数字字符：str[i]>='0' && str[i]<='9'即为数字字符，可以将数字字符写入一个新的字符数组中 str1[j]= str[i];
- 查找指定字符第一次出现的位置：找到该字符即退出查找（break），可以向处理素数一样，设置标志变量，如果找到 flag=1；未找到 flag=0；最后通过对 flag 的值进行判断即可知道结果。