

不可思议的C++复习资料

第一章	数据类型、变量与表达式.....	1
第二章	选择结构.....	10
第三章	函数与重载.....	17
第四章	循环控制.....	25
第五章	数组.....	37
第六章	附录.....	56

第一章 数据类型、变量与表达式

第2题

```
int a=1, b=2, m=1,n=1;
(m=a>b)&&(n=c>d);
cout<<m<<n;
则m、n的值为
```

答案: m=0,n=1

解析: 对于逻辑表达式, 逻辑运算符具有短路特性, 对于a&&b, 只有在a为真的时候, 才判别b的值, 前一个表达式判断为假, 因此后一个表达式不运行。

第3题

```
int a=65;
char b=65;
cout<<a<<"\n";
cout<<b<<"\n";
则输出结果为
```

答案:

65

A

解析: 前一个为int型, 直接输出数据, 后一个为char型, 输出字符常量

第4题

“A”输出___个字符，占有___个字节
‘A’输出___个字符，占有___个字节

答案：

1, 2

1, 1

解析：字符常量占一个字节，字符串的字符序列最后总是添加有一个结束标志\0，占有一个字节。

第5题

```
#include <iostream>
using namespace std;
int main()
{char small='a'; //定义变量赋初值，实际存97
char big;
big=small-32; //字符可参与计算
cout<
```

答案：

A

65

65

解析：第一行输出的是“big”，而big被定义为“char”，这种定义使得在输出“big”时，自动将ASCII码转化为对应字符，故第一行输出的是大写的A；

第二行输出的是“small-32”此处的small需要先进行运算，由“char”的定义可知，此时的“small”已经是一个数字（ASCII码），它参与运算后得出了65，而这个65并非定义为“char”的“small”，因此输出时也不再输出字符，而是一个数字。

第6题

```
#include <iostream>
using namespace std;
int main()
{ int number=123;
int digit,tens, hundreds;
digit=number%10; //功能是什么？
tens=(number/10)%10; // 红线功能是什么？
hundreds=(number/100)%10; //功能是什么？
cout<< hundreds<<tens<<digit<<endl; //输出结果多少？
return 0;
}
```

答案：123

解析：①number/10意味着整除十取商，即去掉个位这个数所得的少一位的数。；

②number%10意味着整除十取余数，即取个位数；

第7题

```
#include <iostream>
using namespace std;
int main()
{
```

```
int a=12;
a+=a-=a*a;
cout<
```

答案： -264

解析：从右往左运算，先算 $a*a=144$ ，然后 $a=a-144$ ，即将 $12-144$ 的结果赋给 a ，故 $a=-132$ ，最后是 $a=a+(-132)$ ，此时右边的 a 已经被赋了 -132 ，故新的 a 将被赋 -264 ，随后输出。

第8题

```
#include <iostream>
using namespace std;
int main()
{ int a=5,b=7,x,y;
  x=a+(b++);
  y=(++a)+b;
  cout<
```

答案： 681214

解析： $x=5+7, b=b+1, a=a+1, y=a+b$;故 $a=6, b=8, x=12, y=14$

第9题

```
int a,b;double d=5.4;
```

表达式 $d*(b=((a=2.1)+1.5)/6.0)$ 的值为 **(0.0)**。

解析：这道题的考点在于计算和赋值时发生的隐式转换。按照运算顺序，由于 a 是整型变量，表达式 $a=2.1$ 的值为 2 ，则 $b=(2+1.5)/6.0$ ；由于 b 也是整型变量，则该表达式值也为 0 ，所以表达式等价于 $d*0$ ，而 d 为 $double$ 型变量，则结果也为 $double$ 型变量，。应写作 0.0 。

第10题

```
int a=3;
a+=a-=a*a;
a=? (B)
```

A: 12

B: -12

C: -3

D: 9

答案： B

解析： $+=$ 、 $-=$ 相加后赋值，相减后赋值。

从右往左计算并赋值

第一步：（此时 $a=3$ ） $a-a*a=a=-6$

第二步：（此时 $a=-6$ ） $a+a=a=-12$

$a+=a == a=a+a$

$a-=b == a=a-b$

从左至右计算，从右至左赋值

解析补充：

C++运算符优先级：

乘法优先级高，所以先算乘法 $a*a=9$ 此时 a 的值没变，还是 3

然后计算 $a+=a-=9$

因为 $+=$ 和 $-=$ 的结合性是从右向左 相当于

$a+=(a-=9)$

$a+=-6$ //此时 $a=-6$

所以最后 $a=-12$

第11题

若 a 是float型变量，则表达式 $a=1,a+1,a+=a$ 的值为【】。

答案:2

解析：中间表达式没有改变 a 的值，第三个表达式的意思是： $a+a$ 得到一个值，然后把值赋给 a ，因此为2

第12题

已知 $\text{int } i=0, j=1, k=2$; 则逻辑表达式 $++i||-j\&\&++k$ 的值为【】

解析： $++i||-j\&\&++k$ 等价于 $++i||(-j\&\&++k)$ ，先计算 $++i$ ，短路特性后面的运算不进行了，最后 $i=1, j=1, k=2$

第13题

$\text{int } i=5, j=6, a=12, x, y;$

$x=i++;$ //后置自增，先将 i 的值赋给 x ，再使 $i=5+1=6$

$y=-j;$ //前置自减，先使 $j=6-1=5$ ，再将 j 的值赋给 y

$x=j++;$ // $x=5$ j 等于 $5+1=6$

$a+=a-a*a;$ //赋值运算符从右向左进行计算 $a*a=144$ $a=a-a*a=-132$; $a=a+a=-264$

i, j, a, x, y 的值?

故所求值分别为 6 6 -264 5 5

第14题

编写程序，输入摄氏温度 C ，输出华氏温度 F 。其中： $F=9/5C+32$ 例如：人体温度摄氏 37°C 用华氏温度表示就是： $F=9/5*37+32=98.6$ 度。所以在西方国家看病医生说你的体温是99度，千万不要吃惊，他说的是华氏温度。

输入格式:

在一行中输入摄氏温度。

输出格式:

对每一组输入，输出其华氏温度 F 。

输入样例:

在这里给出一组输入。例如:

37

输出样例:

在这里给出相应的输出。例如:

$F=98.6$

编译如下: (C++)

```
#include <iostream>
using namespace std;
int main()
{
    double c,f;
```

```

    cin>>c;
    f=9.0/5.0*c+32;
    cout<<"F="<<f<<endl;
    return 0;
}

```

启示：整型除以整型结果默认为整型，如需输出为浮点数，则应改为浮点数除以浮点数。

第15题

编写程序，读入一个正3位整数，然后输出按位逆序的数字。注意：当输入的数字含有结尾的0时，输出不应带有前导的0。比如输入700，输出应该是7。记录程序代码及其运行结果。

输入格式：

在一行中输入3位正整数。

输出格式：

对每一组输入，输出逆序数。

```

#include<iostream>
using namespace std;
int main()
{
    int a,b,c,x;
    cin>>x;
    a=x/100;
    b=x/10%10;
    c=x%10;
    cout<<c*100+b*10+a<<endl;
    return 0;
}

```

第16题

编写程序，键盘输入圆柱体的底面半径和高，计算并输出圆柱体的表面积和体积（圆周率按3.14计）。

输入格式：

在一行中输入圆柱体的底面半径和高。

输出格式：

对每一组输入，在兩行中分别输出表面积S和体积V。

```

#include<iostream>
using namespace std;
#define PI 3.14
int main()
{
    double r,h,S,V;
    cin>>r>>h;
    S=2*PI*r*r+2*PI*r*h;

```

```

V=PI*r*r*h;
cout<<"S="<<S<<endl;
cout<<"V="<<V<<endl;
return 0;
}

```

第17题

已有定义：int x=3,y=4,z=5;则表达式 $!(x+y)+z-5\&\&(y=z/2)$ 的值是 ()

答案：0

此题需注意，逻辑反运算符反的只是 $(x+y)$ ，而不是左侧表达式整体。

第18题

已有定义：int x=3,y=4,z=5;则表达式 $!(x+y)+z-5\&\&(y=z/2)$ 中y的值是 ()

答案：4

此题需注意，逻辑与运算符当第一个表达式为否时(即0)则运算整体为0，故而运算符右边的表达式不会被执行($\&\&$ 是短路运算符)，当第一个表达式为真时，右侧的表达式才会被执行，故此题y的值没有变化

同时此题还需特别注意认真审题，看清楚到底所求是y的值还是逻辑运算的值。

第19题

逻辑非运算符、关系运算符、算术运算符、赋值运算符、逻辑与或运算符的优先级

答案：

逻辑非运算符>算术运算符>关系运算符>逻辑与或运算符>赋值运算符

解析：

见优先表

第20题

以下两个程序段输出结果。(条件表达式，若同级别，从右往左结合)

```

void main()
{   int a=10,b=20,d=15,c;
    c=a>b?a:b>d?b++:d++;
    cout<<c<<','<<b<<','<<d<<endl;
}

void main()
{   int a=10,b=20,d=15,f;
    f=a<b?a:b>d?b++:d++;
    cout<<f<<','<<b<<','<<d<<endl;
}

```

答案：20， 21， 15

10， 20， 15

解析： $c=a>b?a:b>d?b++:d++;$

相当于 $c=a>b?a:(b>d?b++:d++)$ ，即 $c=a>b?a:X$ 。先计算 $a>b$ 结果为假，则取X部分，即计算 $c=b>d?b++:d++$ 。因 $b>d$ 为真，则取 $c=b++$ ，因后缀，所以b的原值20赋给c，b再自增1。

第二段 $f=a<b?a:b>d?b++:d++$

相当于 $f=a<b?a:(b>d?b++:d++)$ ，即 $f=a<b?a:X$ 。先计算 $a<b$ 结果为真，取a即可，不会计算X的(只有 $a<b$ 结果为假才计算X)

第21题

从键盘输入三个小写字母，输出其 ASCII 码值，以及大写字母

```
#include<iostream>
using namespace std;
int main(){
    char a,b,c;
    int d,e,f;
    cin>>a>>b>>c;
    d=a;
    e=b;
    f=c;
    a=a-32;
    b=b-32;
    c=c-32;
    cout<<a<<" "<<b<<" "<<c<<" "<<d<<" "<<e<<" "<<f;
    return 0;
}
```

第22题

下面的程序输出的结果为 ()

```
int a=3;
void main()
{
    int s=0;
    {int a=5;s+=a++;}
    s+=a++;
    cout<<s<<endl;
}
```

答案：8

解析：第5行局部优先，a的值为5，s=0+5=5，第6行为全局变量，a的值为3，s=5+3+8。注意后缀自增是在a参与完美表达式的运算之后再自增的。前缀自增是在参加表达式运算之前先自增的。

第23题

设有int x=11; 则表达式(x++*1/3)的值为

答案：3

解析：x的原始值参与表达式运行后，x的值再加一，即 $11 \times 1/3 = 3$

第24题

若变量 a 是 int 类型，执行了语句 a='A'+1.6;，则下面说法中正确的是【 】。

- A) a 的值是字符 C
- B) a 的值是浮点型
- C) 不允许字符型和浮点型相加
- D) a 的值是字符'A'的 ASCII 值加上 1

答案：D

解析：字符“A”要转换成相应的ASCII码值，由于运算结果要赋值给int型变量，所以对 1.6 进行取整运算，最后a的值应是66。

第25题

下面程序的输出结果是【A】

```
#include<iostream>
using namespace std;
int a=3;
int main()
{
    int s=0;
    {
        int a=5;
        s+=a++;
    }
    s+=a++;
    cout<<s<<endl;
    return 0;
}
```

- A、 8
- B、 10
- C、 7
- D、 1

解析：s+=a++即：s=s+a, a=a+1, 因此在第一个“s+=a++”中，s=5,a=6，但由于上述运算在花括号内，a的自增并未生效，第二个“s+=a++”中，s=s+a=5+3=8。

温馨提示：i++和++i的区别：

- 1)、 a = i++;
相当于 a = i; i++;
- 2)、 a = ++i;
相当于 i++; a = i;

第26题

下面程序段的输出结果是【 】。

```
int i=011,j=11;
cout<<++i<<","<<j--<<endl;
```

答案：10,11

解析：i为八进制数，转换为十进制数时,i=9,输出时，先将表达式计算完再输出，故输出结果为10,11

第27题

以下语句执行后：int x=2; double y; y=(int)(float)x;，变量 x 和 y 的数据类型分别是

答案：整型、整型

解析：y被强行转换为int型数据，而把数据x用在数据转换的过程中对数据x的原来类型无影响。

第28题

“\72”在内存中占【 】个字节，“\72”在内存中占【 】个字节。

答案：1, 2

解析：\72表示一个字符：， 因此占一个字节，“\72”=“\72”+“\0”，因此占两个字节。

第29题

已知'0'的 ASCII 码为 48，下面程序段的输出结果是【 】。

```
char c1='6',c2='0';
cout<<c1<<','<<c2<<','<<c1-c2<<','<<c1+c2;
```


答案: 6,0,6,102

解析: 第一、二个输出的数据为字符型数据, 第三、四个输出的数据为整型数据。对于第三、四个数据, 若要输出字符型数据, 需先定义一个字符型变量接收运算结果, 后输出该字符型数据, 否则, 系统会自动输出其ASCII值运算后的结果。

第30题

下列程序的输出结果是【 】。

```
#include <iostream>
using namespace std;
int main()
{
    int a=5;
    cout<<sizeof(a)<<','<<sizeof(int)<<endl;
    double b=1.5;
    cout<<sizeof(1.5f)<<','<<sizeof(b)<<','<<sizeof(1.5L)<<endl;
    return 0;
}
```

答案:

4,4

4,8,16

解析: int型数据占4个字节, float型数据占4个字节, double 型数据占8个字节, Long double 型数据占16个字节。

第31题

下面程序段的运行结果是【 】。

```
if(2*2 == 5<2*2 == 4)
    printf("T");
else printf("F");
```

答案: F

解析: 因为<优先级高于==, 此时先判断5<2*2, 为0, 此时, if内变为, 2*2==0==4, 再从左至右判断, 为0, 即输出F

第32题

3.输出两个标准格式时间(hh:mm:ss)之间的秒数差。输入格式: 包括两行, 两行均为“hh:mm:ss”格式的时间。且题目保证第一个时间不大于第二个时间

```
#include<iostream>
using namespace std;
int main(){
    int h,m,s,t1,t2;
    char a;
    cin>>h>>a>>m>>a>>s;
    t1=h*3600+m*60+s;
    cin>>h>>a>>m>>a>>s;
    t2=h*3600+m*60+s;
    cout<<(t2-t1);
```

}

解析：题目主要考查两点：一是对于时、分、秒之间的六十进制转换；二是对输入数据之间的字符（即冒号“:”）的处理，需要用字符变量来接收、存储，而不能认为可以直接连续用三个整型变量存储三个整数。

第33题

当int a=3,b=2,c=1,f 时,表达式f=a>b>c的值是

答案：0

解析：先判断a>b，为真，再判断1>c为假，则f=0

```
int a;

int main()
{
    a=5;
    cout<<"global a="<<a<<endl;
    {
        int a=10;
        cout<<"local a="<<a<<endl;
        a++;
        cout<<"local a="<<a<<endl;
    }
    cout<<"global a="<<a;
}
```

解答：

global a=5

local a=10

local a=11

global a=5

第二章 选择结构

第35题

判断输入的三个正整数是否相邻，例如14,13,15相邻则输出“TRUE”，否则输出“FALSE”。

思路1:排序后判断

```
#include<iostream>
using namespace std;
int main(){
    int a,b,c,t;
    cin>>a>>b>>c;
    if(a>b){t=a;a=b;b=t;}
    if(a>c){t=a;a=c;c=t;}
    if(b>c){t=b;b=c;c=t;}
    if(b-a==1&& c-b==1) cout<<"TRUE";
```

```

        else cout<<"FALSE";
        return 0;
    }

```

清新脱俗的思路2:由三个相邻数的六种排列组合情况分类讨论

```

#include<iostream>
using namespace std;
int main(){
    int a,b,c;
    cin>>a>>b>>c;
    if(a-b==1||b-a==1)//若a、b相邻则c+1=较小数或c-1=较大数
        if(c+1==(a<b?a:b)||c-1==(a>b?a:b)) cout<<"TRUE";//注意三目运算符优先级较低
        else cout<<"FALSE";
    else if(a-b==2||b-a==2)//若a、b相差2,则c必为其平均数
        if(c*2==a+b) cout<<"TRUE";
        else cout<<"FALSE";
    else cout<<"FALSE";
    return 0;
}

```

第36题

```

int x=9;
if (x==0) cout<<"test 1 ok\n";
if (x=5) cout<<"test 2 ok\n";
if (x=0) cout<<"test 3 ok\n";

```

则输出结果为

答案: test 2 ok

解析: 对if(x==0)而言, x不等于0, 不运行; 对if(x=5)而言, 即为if(true), x=9!=0, 因此符合条件, 对if(x=0)而言, 即为if(false), x不等于0, 不运行。

第37题

```

char c='a';
if('0'<=c<='9') cout<<"YES";
else cout<<"NO";

```

程序段的运行结果为 (YES)

解析: 这道题强调了程序中表达式往往区别于数学表达式。'0'<=c<='9'的正确运算顺序为('0'<=c)<='9'.其值为1, 因此输出"YES"。这个程序极容易被误解为"判断变量c的值是否在'0'与'9'之间", 而实际上要达到该目的所使用的判断条件应表述为'0'<=c&& c<='9'.在编程时应当提高警惕, 避免混淆数学与编程的运算。

第38题

下面程序段执行后, x、y、z的值是 【 】。

```

int x=10,y=20,z=30;
if(x>y)
    z=x; x=y; y=z;
cout<<x<<","<<y<<","<<z;

```


- A、 20,30,10
- B、 10, 20,30
- C、 20,30,30
- D、 20,30,20

答案： C

易错答案： B

错误原因： 很多人都容易选择b，原因是在没有大括号的前提下if不成立是只不执行if后的一条语句，而继续执行其他的语句，而非不执行所有语句（出题者常常把语句放到一行上迷惑人）。

第39题

阅读下面的程序,写出程序的输出结果。

```
#include<iostream>
using namespace std;
int main( )
{
int a, b, c;
a=2;b=3;c=1;
if(a>b)
if(a>c)
cout<<" a:"<<a;
else
cout<<" b:"<<b;
cout<<" c:"<<c;
return 0;
}
```

答案： c:3

解析： 本题考查的点在于在没有大括号时具有多个if， else语句时如何去“断句”，只需要把握else只与它最近未配对的if相配对这个原则，即可。

第40题

```
{ float score;
cout<<("输入学生成绩(0~100):");
cin>>score;
if(score>100||score<0)
cout<<("输入分值错误! ");
else
switch( 【1】 )
{ case 10:
case 9:cout<<"优";break;
case 8: cout<<"良"; break;
case 7: cout<<"中"; break;
case 6: cout<<"及格"; break;
default: cout<<"不及格";
}
```

```
}
```

答案： (int) score/10

解析：这道题由于定义的变量为float型，而switch后面的只能是整形和浮点型，所以一定要进行类型的转换。

第41题

下面的程序，是否有错，请指出。

```
void main()
{ int a,b;
cin>>a>>b;
if(a>b)
a=b;b=a;
else
a++;b++;
```

答案： 若if—else之间有多条语句，必加大括号，否则编译错误。

第42题

编写程序，输入3个整数，判断并输出以这3个整数作为边长能否构建1个有效的三角形。若三个边长为a、b、c，则它们能构成有效三角形需同时满足：

输入格式：

输入在一行中输入3个整数。

输出格式：

在一行输出是否构成三角形(This is a triangle. 或者 This is not a triangle.)

输入样例：

```
#include<iostream>
using namespace std;
int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    if(a>0&&b>0&&c>0&&(a+b>c)&&(a+c>b)&&(b+c>a))
        cout<<"This is a triangle."<<endl;
    else
        cout<<"This is not a triangle."<<endl;
    return 0;
}
```

第43题

a, b, c为整型，以下语句执行后，a中的值是____； b中的值是____； c中的值是____。

```
a=10; b=20; c=30;
```

```
if(a>b) a=b; b=c; c=a;
```

答案： 10 30 10

If () “ () ”中的条件语句成立时，执行下一条语句，不成立时，跳过后面的语句。

第44题

```
int a,b,c;
a=10; b=20; c=30;
if(a>b)
```

a=b;
b=c;
c=a;
a,b,c的值?

答案: 10, 30, 10

此题需注意if到a=b; 算一句, 后面的也要继续运行。

第45题

下列程序段输出结果? 须解析过程。

```
int main()
{ int a = 0, b = 0, c = 0, x = 35;
  if (!a) x--;
  else if (b);
  if (c) x = 3;
  else x = 4;
  cout<<x;
}
```

答案: 4

解析: 第一个if判断, 非a即为非0, 判断为真, 执行"x--", x变为34,同时"else if (b)"被跳过。随后判断if (c), c为零, 判断为假, 执行else x=4。最后输出得4

第46题

在下列条件语句中, 只有一个语句在功能上与其它三个语句不等价 (其中s1、s2表示某条语句), 这个不等价的语句是 ()。

(5.0分)

- A、if(a==1) s1; else s2;
- B、if(!a) s2; else s1;
- C、if(a==0) s2; else s1;
- D、if(a) s1; else s2;

答案: A

解析: A选项只有当A取值1时, 才能输出s1; 其余选项当A不等于0时均可输出s1

第47题

下面程序段输出是【】。

```
int x=10,y=20,z=30;
if(x>y)
    z=x; x=y; y=z;
cout<<x<<","<<y<<","<<z;
```

(5.0分)

- A、20,30,10
- B、10, 20,30
- C、20,30,30
- D、20,30,20

答案: C

解析: if语句只执行下一个分号之前或者后一个配对else中的语句, 即此程序if(x>y) z=x;执行完毕后便直接开始赋值语句, x=20,y=30

第48题

假设有如下的程序片段，若grade的值为'C'，则输出结果是【】。

```
switch(grade)
{
    case 'A': cout << "GREAT!";
    case 'B': cout << "GOOD!";
    case 'C': cout << "OK!";
    case 'D': cout << "NO!";
    default : cout << "ERROR!";
}
```

- A、不确定
- B、 GREAT!GOOD!OK!NO!ERROR!
- C、 OK!NO!ERROR!
- D、 OK!

答案：C

解析：在switch语句中，case后面加break;才能直接到花括号处，而本题的case后未加break;则继续翻译语句直至花括号处。

第49题

找错误

```
Int main()
{ int x,y; cin>>x;
  if (x>=0) ;
  if(x<=10) y=2*x;
  else y=x;
  else y=0;
  cout<<y;
}
```

解析：一定要注意；的使用与误用。
去掉第一个if () 后的分号。

第50题

```
#include<iostream>
using namespace std;
#define N1 10+5
int main()
{
    const int N2=10+5;
    cout<<N1<<" "<<N2<<endl;
    cout<<N1*2<<" "<<N2*2<<endl;
    return 0;
}
```

程序运行结果是：

答案：

15,15

20,30

解析:

第一行, $N1=10+5=15$, $N2=15$

第二行, $N1=10+5*2=20$, $N2=15*2=30$

对于#define

#define末尾没有分号, 如

#define N 5; 会用5;代替N, 而不是5代替N

仅仅是字符替换, 不会做任何转换, 如:

#define X 2+3

用2+3替代X, 而不是用5替代X

对于const

const 后的表达式会先进行计算, 在赋值。

第51题

下列两个switch语句, 你认为结果是相同还是不同?

A.

```
switch(x/10)
```

```
{
```

```
case 0: cout<<"1"; break;
```

```
case 1: case 2: case 3: cout<<"3"; break;
```

```
default: cout<<"4";
```

```
}
```

B.

```
switch(x/10)
```

```
{
```

```
case 0: cout<<"1"; break;
```

```
case 1||2||3: cout<<"3"; break;
```

```
default: cout<<"4";
```

```
}
```

答案:

不同

解析:

对于B语句, $1||2||3$ 的值恒为真, `case 1||2||3: cout<<"3"; break;` 语句一定执行

第52题

输入一个同学的成绩, 输出相应的等级A-E

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{ int score;
```

```
cin>>score; //输入100以内分数
```

```
switch(【 1 】)
```

```
{ case 100:
```

```
case 90: cout<<"A\n"; break;
```

```
case 80: cout<<"B\n"; break;
```

```
case 70: cout<<"C\n"; break;
```

```
case 60: cout<<"D\n"; break;
```

```
//以下不及格输出E
```

```
    【 2 】 : cout<<"E\n";  
    return 0;  
}
```

第一空: `score/10*10`

第二空: `default`

提示: 此题第一空容易误想为用不等式或者其他形式, 但又写不出来, 特选为例题。

第16题当`int a=3,b=2,c=1,f` 时, 表达式`f=a>b>c`的值是【】。

答案: 0

第53题

“水仙花全家福”

判断一个数是否为“水仙花数”, 如果是则输出YES, 否则输出NO。所谓“水仙花数”是指一个3位数, 其各位数字的立方和等于该数本身。例如, 153是一个“水仙花数”, 因为 $153=1^3+5^3+3^3$ 。

样例输入: 1

样例输出: NO

样例输入: 153

样例输出: YES

```
#include<iostream.h>  
#include<math.h>  
int main()  
{  
    int g,s,b,n;  
    cin>>n;  
    g=n%10;  
    s=n/10%10;  
    b=n/100%10;  
    if(n==g*g*g+s*s*s+b*b*b)  
        cout<<" YES";  
    else cout<<" NO";  
}
```

第三章 函数与重载

第54题

以下函数定义中能正确编译的是

- A. `double fun(int x,int y) {int z; z=x+y; return z;}`
- B. `double fun(int x,int y) { z=x+y; return z;}`
- C. `double fun(int x,int y) {int x,y; double z; z=x+y; return z;}`
- D. `double fun(int x, y) {int z=x+y; return z;}`

答案: A。

解析：B项没有定义z；C项重复定义了xy；D项函数定义的时候，必须指明所有函数的类型，所以int x, y不正确。

第55题

下面不正确的说法是：C++语言规定 【】。

- A.实参可以是常量、变量或表达式
- B.实参传递给形参时,要求实参必须有确定的值
- C.形参与其对应的实参类型可以不一致
- D.形参可以是常量、变量或表达式

答案：D

【详解】：形参只能为变量，不可以是表达式或常量。

实参可以是常量、变量、表达式、函数等任意类型，但形参只能是变量，在被定义的函数中，必须指定形参的类型。

【相关知识点】

形式参数（形参）：定义函数名和函数体时需要用的参数，目的是用来接收调用该函数时传递的参数。

实际参数（实参）：传递给被调用函数的值。

通俗的来讲，形参就像物理计算中的公式中的符号，比如 Ω ，A，V等，而实参就是在计算中的实际的数值。调用过程就是实现形参与实参的结合，把实参的值通过调用传递给形参，相当于把具体的值代入公式进行计算。

第56题

下列函数调用中，不正确的是 【】。

- A.max(3,a+b);
- B. max(3,5);
- C.max(a,b);
- D.int max(a,b);

答案：D

【详解】应该是int （max(a,b)），定义变量一定是一个整体。

第57题

有如下程序，编译运行这个程序将出现的情况是（）

```
#include<iostream>
using namespace std;
void function(double val);
int main()
{
    double val;
    function(val);
    cout<<val;
    return 0;
}
void function(double val)
{
    val=3;
}
```

- A.输出一个不确定的数
- B.输出:3
- C. 编译出错,无法运行
- D.输出:3.0

答案: A

【分析】我的观点1、给形参赋值跟实参没关系,而主函数没给实参val赋值,所以输出值不确定; 2、不同数据类型可以赋值只是会强制转换数据类型。

第58题

C++语言中,函数直接或间接调用函数本身,则该函数称为【 】函数。

答案 (1) 递归

第59题

找出以下函数声明的错误。

(提示: 声明须给出形参的个数和类型,及返回值类型,形参名可不写)

```
int fun(int x,int y); //声明1
int lsmin(int,int,int); //声明2
bool lsleap(int y); //声明3
void swap( int x; int y); //声明4
```

答案:

- 1: y需要int
- 2: 声明需要分号,此处缺少分号
- 3: 正确
- 4: 形参之间逗号隔开

函数定义易错点,请找出错误。

问1) 定义函数时后面加分号对不对?

问2) 将形参y挪至{}里面定义行否?

问3)这里出现两个return语句可不可以?

```
bool lsleap(int y); //返回值为bool型
{
    int y;
    cin>>y;
    if(y%4==0&& y%100!=0||y%400==0)
    return true; //true实际为1,false实际为0
    else return false;
}
```

答案:

- 1)不对
- 2)不行
- 3)不行

解析:

定义时不需要加分号(可观察main函数理解),声明时需要加分号。

形参的定义是一种简单明了便于理解的规则,不能打破规则。

3) 作为整个函数,只能有一个返回值,如果这个返回值会随不同的情况而改变,即可能涉及到if语句,就会出现多个return语句。不过因为if语句只能选择一个执行,所以返回值只有一个。

第60题

若以下选项中的变量都已正确定义并赋值，则对函数fun的正确调用语句是_C。

A.fun(int y,double m);

B.k=fun(10,12.5);

C.fun(x, n);

D.void fun(n,x);

解析：函数调用的一般形式为：函数名称 （实际参数）

其中实际参数为在主函数中已经定义过的变量，不能直接在调用时赋值。

A调用时不用声明实参类型

B不能在调用时才赋值

D调用时不用标注调用函数类型

第61题

找出以下函数定义的错误，并回答问题。

提问：a.定义函数时，函数头后面能否加分号？

b)函数名能否直接用作变量名？

c) 函数的返回类型是double?由return返回值类型决定？

```
int getAbs(double x); //1)
```

```
{ getAbs=(x<0)?(-x):x; //2)
```

```
return getAbs; //3)
```

答：a.否 b.不能 c.否，否

解析：区别函数的声明和定义的区别，函数的声明要以分号结尾，函数的定义结尾不加分号。函数的返回类型是int，应该由函数类型决定。

第62题

形参的改变是否影响实参？

提问(1)程序段输出结果？

(2)请比较swap前后a, b结果，形参xy的改变是否影响实参ab值？

```
void swap( int x, int y);
```

```
int main()
```

```
{int a=2,b=3;
```

```
cout<<"swap() 前:"<<a<<" "<<b<<endl;
```

```
swap(a,b);
```

```
cout<<"swap() 后:"<<a<<" "<<b<<endl;
```

```
}
```

```
void swap( int x,int y)
```

```
{int t;
```

```
t = x; x = y; y = t;
```

```
cout <<"x="<< x <<" "<<"y="<<y<< endl;
```

```
}
```

答案与解析：输出结果为：

swap()前2,3

x=3,y=2

swap()后： 2,3

不会影响。

调用函数时,实参复制给形参,单向值传递,实参形参各自独占内存

第63题

参数默认值问题

提问(1)找出下列程序段的错误。

(2)main()调用3的实参分别多少?

```
long set4DPoint ( int x, int y, int z, int t); //声明1
long set4DPoint ( int x, int y, int z=1, int t); //声明2
long set4DPoint ( int x, int y=2, int z, int t=2000); //声明3
long set4DPoint ( int x, int y, int z=1, int t=2000); //声明4
long set4DPoint ( int x, int y, int z=1, int t=2000) //函数定义
{.....}
main()
{ cout<<set4DPoint (2,4,6,3000); //调用1
  cout<<set4DPoint (2,4,5); //调用2
  cout<<set4DPoint (2,4); //调用3
}
```

答案: 1、声明2、指定形参默认值,从右向左。所以int t这里错了,应该也指定默认值。

2、声明3, int y=2的右边所有形参都必须指定默认值。

3、声明中指定了,函数定义不用重复指定。

解析:默认参数:若在函数声明中指定了形参默认值,调用函数时可不传递实参值,而使用默认值。

1) 只在函数声明时指定形参默认值,函数定义时不能指明形参默认值。

2) 遵循规则:若某形参指定默认值,其右侧所有形参都必须指定默认值,即从右往左连续给默认值,否则出错

第64题

```
int fun(int a, int b, int c)
{a=456; b=567; c=678;} //形参值改变不影响实参
int main(){
  int a=10, b=20, c=30; //不同函数内可以定义相同变量名,其存储空间相互独立
  fun(a, b, c); //函数调用时实参的值单向传递给形参
  cout<<a<<b<<c;
  return 0;}

```

此程序段的输出结果是(102030)

第65题

有如下函数调用

fun(a+b, 3, max(n-1, b))

则此fun函数的实参个数为(3)

解析:函数调用可以嵌套而函数定义不可嵌套,此外,把a+b,3,max(n-1,b)分别看成3个整体,因此实参的个数为3。

第66题

下面函数原型声明形式中,正确的是 ()

int fun(int x;int y)

int fun(int x int y)

int fun(int x,y)

```
int fun(int x,int y);
```

答案：D

函数声明是一条语句，结尾须用分号，括号里每个参数都得说明类型 故选d

第67题

下面函数声明中，（）是“void BC(int a,int b);”的重载函数

```
int BC(int a,int b)
```

```
void BC(int x,int y);
```

```
double BC(int a,int b,int c=0);
```

```
void BC(double a,double b);
```

答案：D

解析：重载函数是和原函数参数个数或类型不一样，重载函数和原函数相比是参数类型或参数个数不一样

第68题

下面函数调用的语句中，一定不正确的是（）；

A. max (3, a+b) ;

B. max (3, 5) ;

C. max (a,b);

D. max (int a,int b);

答案：D

解析：我们说进行函数调用的时候必须是不带数据类型的，要么是变量，或者带着变量的表达式，要么是常量，或者是函数的调用它的返回值

第69题

编写函数getArea，传入三角形的三条边长作为参数，利用海伦公式计算三角形的面积（保留小数点后2位）。【海伦公式】若三角形的三个边长为a、b、c，设 $s=(a+b+c)/2$ ，则三角形面积为： $area = \sqrt{s(s-a)(s-b)(s-c)}$ 。

编写函数validate，传入3 个边长作为参数，判断是否构成有效三角形，若能构成有效三角形，函数返回true，否则返回false。

```
#include<iostream>
#include<math.h>
#include<iomanip>
using namespace std;
bool validate(double a,double b,double c);
double getArea(double a,double b,double c);
int main(){
    double a,b,c;
    cin>>a>>b>>c;
    if(validate(a,b,c)==false)
        cout<<"Invalid Triangle";
    else
        cout<<"The area is:"<<fixed<<setprecision(2)<< getArea(a,b,c);
    return 0;
}
bool validate(double a,double b,double c)
{bool validate;
```



```

    if(a+b<=c||a+c<=b||b+c<=a)
    return 0;
    else
    return 1;
}
double getArea(double a,double b,double c)
{
    double s,area;
    s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    return area;
}

```

第70题

以下叙述中不正确的是【D】

- A、函数中的形式参数是局部变量
- B、在不同的函数中可以使用相同名字的变量
- C、在一个函数内定义的变量只在本函数范围内有效
- D、在一个函数内的复合语句中定义的变量在本函数范围内有效

解析：在函数定义时声明的参数只在函数内部起作用，不会影响到其他函数，是函数的局部变量，A)正确。在不同的函数中使用相同的函数名不代表是同一函数，B)正确。在一个函数中定义的变量是这个函数的局部变量，所以只在这个函数内起作用，C)正确。复合语句中定义的变量其作用域是这个复合语句，不会扩大到整个函数，所以D)错误。

第71题

若变量a,b有确定的值，则下列函数调用中，不正确的是【D】。

- A、max(3,a+b);
- B、max(3,5);
- C、max(a,b);
- D、max(int a,int b);

解析：C语言中，函数调用的一般形式为：函数名(实际参数表)。对无参函数调用时无实际参数表。实际参数表中的参数可以是常数、变量或其它构造类型数据及表达式。各实参之间用逗号分隔。

第72题

下面程序的输出结果是【B】

```

#include<iostream>
using namespace std;
int fun(int a=1,int b=2,int c=3);
int main()
{
    cout<<fun(4);
    return 0;
}
int fun(int a,int b,int c)
{
    return a*b*c;
}

```


}

- A、 6
- B、 24
- C、 12
- D、 8

解析：函数的初值为int a=1,int b=2,int c=3, fun(4)将a的初值改变为4, 根据函数定义我们可知, fun(4)即为 $4*3*2=24$ 。函数默认值要从右到左赋值; 函数调用时, 从左到右进行赋值。默认值在调用的时候可以改变。

第73题

下面不正确的说法是: C++语言规定 **【D】**

- A、 实参可以是常量、变量或表达式
- B、 实参传递给形参时,要求实参必须有确定的值
- C、 形参与其对应的实参类型可以不一致
- D、 形参可以是常量、变量或表达式

解析：形式参数（形参）不可以是常量, 可以是变量, 通常也不可以是表达式, 但可以是某些特殊的表达式。如: a[]; 实参与形参的类型应相同或赋值兼容

第74题

下面关于形参与实参的说法中, 不正确的是 **【B】**

- A、 形参是函数的局部变量, 函数调用完毕后, 局部变量就被销毁
- B、 形参与实参如果都是变量, 则变量名称与数据类型必须完全相同
- C、 实参可以是常量、变量、表达式或其它函数调用的返回值
- D、 函数调用时将实参的值传递给形参, 函数调用结束后并不能将形参的值传递给实参

解析：实参向形参的数据传递是“值传递”, 单向传递, 只能由实参传给形参, 而不能由形参传给实参。实参和形参在内存中占有不同的存储单元, 实参无法得到形参的值。在调用结束后, 形参单元释放。实参可以是常量、变量、表达式, 实参与形参的类型应相同或赋值兼容。

第75题

题目：有5个人坐在一起, 问第五个人多少岁? 他说比第4个人大2岁。问第4个人岁数, 他说比第3个人大2岁。问第三个人, 又说比第2人大两岁。问第2个人, 说比第一个人大两岁。最后问第一个人, 他说是10岁。请问第五个人多大?

```
#include<iostream>
using namespace std;
int age(int n){
    int c;
    if(n==1) c=10;
    else c=age(n-1)+2;
    return c;
}
int main(){
    cout<<age(5);
    return 0;
}
```

第76题

下列程序的输出结果是 **【 】**。

```

int x;
void cude()
{ x=x*x*x; }
int main()
{ x=5;
  cude();
  cout<<x;
  return 0;
}

```

答案：125

解析：x为全局变量，改变任意一个x的赋值，x的输出值都会改变。

第四章 循环控制

第77题

从键盘输入N，输出1到N；

```

#include<iostream>
using namespace std;
int main(){
  int i,N;
  cin>>N;
  i=1;
  cout<<i<<endl;
  while(i<N){
    i++;
    cout<<i<<endl;
  }
  return 0;
}

```

第78题

将三个不相等的整数按从小到大的顺序和从大到小的顺序排列

三个不相等整数从键盘输入

```

#include<iostream>
using namespace std;
int main(){
  int a,b,c,d;
  cin>>a>>b>>c;
  if(a>b){
    d=a;
    a=b;
    b=d;}
  if(a>c){
    d=a;

```

```

        a=c;
        c=d;
    }
    if(b>c){
        d=b;
        b=c;
        c=d;
    }

    cout<<a<<b<<c<<endl;
    cout<<c<<b<<a<<endl;
    return 0;
}

```

第79题

改造题目，将不相等去掉，如何改写上面的程序依然可行

```

//sort
#include<iostream>
#include<algorithm>
using namespace std;
int main(){
    int a[3];
    for(int i=0;i<3;++i) cin>>a[i];
    sort(a,a+3);
    for(auto e:a) cout<<e<<" ";
    return 0;}

```

第80题

- 1) A中for循环的i++每次循环都执行否？
- 2) B中若continue;执行,i值是否会变？导致什么结果？
- 3) 程序段B的循环能否正常结束？

/* 程序段A*/

```

s=0;
for(i=1;i<=10; i++) //i++每次循环都执行否？
{ if(i%2==0)
    {continue;}
  s=s+i;
}

```

/* 程序段B */

```

s=0;
i=1;
while(i<=10)
{ if(i%2==0)
  {continue;} //若continue执行,i值是否会变？
  s=s+i;
  i++; //此句每次循环都执行否？
}

```



```
}
```

答案:

- 1) 每次都执行。
- 2) i值不会变, 导致死循环。
- 3) 不能 (只有第一次循环执行了)。

解析: 1) continue的作用是终止本次循环。执行下一次for循环前, 会先更新循环控制变量i, 再检测布尔表达式i<=10。所以i++每次循环都执行

2) 初始i=1, 可知i%2==0为假, 进行"s=s+i;i++"后i=2, 判断为真, 不再执行余下语句, 继续下一次判断, 进入死循环。

3) i永远为2

第81题

分析程序段功能, 回答问题。

提问: 1) continue执行时, i值多少? i值还会再变么? while是否进入死循环?

2) 循环过程中, 怎样才能避免进入死循环?

```
s=0;
i=1;
while(i<=10)
{ if(i%2==0)
{continue;}
s=s+i;
i++;
}
```

答案和解答:

- 1) i值为2, 之后i值不会改变, 进入死循环。
- 2) 循环过程应该保证循环变量的值在运行过程中可以得到修改, 并使循环条件为假, 从而结束循环, 比如使用for语句。

第82题

下列程序段执行后s的值为?

```
int i=5, s=0;
while(i-->0)
{ if(i%2)
continue;
else s+=i;}
```

答案: 6

解析: 因为当表达式i-- 的值为4, 2的时候, i的值继而变为3, 1, continue进入下一个循环, 当表达式i--的值为5, 3, 1的时候, i的值自减变为4, 2, 0, 执行s+=i; 因此s的值为(4+2+0)=6

第83题

对比以下2个程序, 他们的执行结果分别为多少?

程序一:

```
int k=0;
while(k++<=2) cout<<k<<" ";
```

执行结果为: 1 2 3

程序二

```
int
```

```
k=0;
while(k++<=2);
cout<<k;
```

执行结果为：3

解析：对比第二题，第三题两者差别在于while后的分号，如果有分号则循环语句就为空语句，这样每次循环只为一句话，而不执行之后的语句。

(补充：第一个程序输出结果不是“1 2”的原因：k=0,1,2条件为真，每次判断后k自增，对应的输出值为1，2，3。

第84题

下面的程序是用“辗转相除法”求两个整数的最大公约数和最小公倍数,请填空。(注下面代码的预处理部分省略)

```
int main()
{int r,m,n,t;
cin>>m>>n;
if(m<n) //m大n小
{ t=m; m=n; n=t;}
t=m*n;
r=m%n;
while(r!=0)
{m=n; n=r; r= 【 】 ;}
cout<<"最大公约数是:"<<n<<endl;
cout<<"最小公倍数是:"<< 【 】 <<endl;
return 0;
}
```

答案：

(1) m%n

(2) t/n

第85题

下面程序段【死循环】。

```
x=3;
do
{
y=x--;
if(!y) cout<<"#";
}
while(1<=x<=2);
```

解析：while中的语句不是错误的，应该这样理解，从左往右看“1<=x”，这个结果要么为真那么为假，所以为0或者1，由于0，1永远小于2，所以while后面的语句成立，将不断进行循环，因此为死循环。

第86题

计算序列 1+1/3+2/5+3/7+4/9.....前10项的和。

提示：实际序列为1+1/3.0+2/5.0+3/7.0 (注意分母为实数，否则每一项均为0)

提问: 1)填空【1】，填i<=10对不对?(看清sum初值)

2) 填空【2】须注意什么?

3)填空【3】,如何调用子函数?

```
double serial()
```

```
{int i;
```

```
double sum=1;//记得sum赋初值
```

```
for(i=1;【1】;i++)
```

```
{sum=sum+【2】;
```

```
}
```

```
return sum;
```

```
}
```

```
main()
```

```
{【3】;
```

```
}
```

答案与解析:

不对,应为“ $i \leq 9$ ”。sum 的被赋初值1,即代表了第一项。

$i/(2*i+1.0)$ 。即为第i次所需要的加入sum的数字。

`cout<<serial();`即调用函数

提示: $a > b > c$ 解为 $(a > b) > c$, $3 > 2$ 成立,所以 $a > b$ 值为真(即为1),又因为 $1 > 1$ 不成立,所以 $f = 0$ 。

补充: 运算符的优先顺序为:逻辑非运算符>算术运算符>关系运算符>逻辑与或运算符>赋值运算符。

第87题

输出所有大于给定数n的“水仙花数”,n由键盘输入。所谓“水仙花数”是指一个3位数,其各位数字的立方和等于该数本身。例如,153是一个“水仙花数”,因为 $153 = 1^3 + 5^3 + 3^3$ 。

样例输入: 100

样例输出: 153 370 371 407

```
#include <iostream.h>
```

```
int main()
```

```
{int n,g,s,b,i,sum;
```

```
cin>>n;
```

```
for(i=n+1;i<=999;i++)
```

```
{g=i%10;
```

```
s=i/10%10;
```

```
b=i/100%10;
```

```
sum=g*g*g+s*s*s+b*b*b;
```

```
if(sum==i)
```

```
cout<<i<<" ";}
```

```
return 0; }
```

第88题

编写程序求n的阶乘,n由键盘输入。

样例输入: 10

样例输出: 3.6288e+006

```
#include<iostream.h>
```



```

int main()
{
    int i=1,n,k=1;
    cin>>n;
    for(;i<=n;i++)
    {
        k=k*i;
    }
    cout<<' '<<k;
    return 0;
}

```

第89题

若执行下面的程序段，则下面描述中正确的是【D】。

```

int k=5;
while(k=1) k--;

```

- A) while 循环执行四次 B) 循环体执行一次
C) 循环体一次也不执行 D) 死循环

解析：k=1是赋值语句，条件恒为真，死循环

第90题

C 语言中 while 和 do-while 的主要区别是【A】。

- A) do-while 的循环体至少无条件执行一次
B) while 的循环控制条件比 do-while 循环控制条件严格
C) do-while 允许从外部转到循环体内部
D) do-while 的循环体不能是复合语句

解析：while是先判断条件再执行语句；do-while则是先执行一次语句再判断条件，从而决定是否继续再执行循环体中的语句，这正是二者主要区别。

第91题

执行程序段，则下面说法中正确的是【C】。

```

x=-1;
do { x=x*x; } while(!x);

```

- A) 死循环 B) 循环执行 2 次 C) 循环执行 1 次 D) 有语法错误

解析：do-while的功能是先执行一次循环体，再判断条件是否成立，当x=-1时，执行do语句，x=x*x=1，接着执行while后的表达式，其值为0，退出循环，因此循环体只执行了一次。

第93题

下面有关 for 循环的正确描述是【D】。

- A) for 循环只能用于循环次数已经确定的情况
B) for 循环中是先执行循环体语句，后判断表达式
C) 在 for 循环中，不能用 break 语句跳出循环
D) for 循环的循环体可以包含多条语句，但必须用花括号括起来

解析：选项A) for循环可以用于循环次数不确定的情况；选项B) for循环是先判断表达式，后执行循环体语句；选项C) 在for循环中，可以用break语句跳出循环体，从而接着执行循环体后面的语句。

第94题

下面程序的功能是求1000以内的完全数,请填空。(说明:一个数如果恰好等于它的因子之和(除

自身外),则称该数为完全数,例如 $6=1+2+3$,6为完全数)(注下面代码的预处理部分省略)

```
int main(){
    int a,i,m;
    for(a=1;a<=1000;a++){
        m=0;
        for( 【1】 ;i<=a/2;i++)
            if(!(a%i)) m= 【2】 ;
        if(m==a) cout<<setw(4)<<a;
    }
    return 0;
}
```

答案: **i=1;m+i**

解析: 寻找完全数的大体思路可分为三步: 1、遍历从1到1000的整数; 2、对每个整数,判断是否是完全数; 3、输出对应的完全数。其重点在于完全数的判断。依据题意,对于某整数 n 是否是完全数的判断又可细分为4步: a.遍历可能存在该整数因数的区间范围内所有整数; b.判断每个整数是否是 n 的因数; c.将除 n 本身的本身的所有因数加和; d.比较因数之和与 n 本身,判断 n 是否是完全数。

无论是编程题亦或是程序填空题,最重要的是对程序的思路清晰明了,明确每一部分语句的作用,捋清思路后,答案往往可以一目了然

第95题

下面程序的功能是:求Fibonacci(斐波纳契)数列的前40个数,并按照4列一行输出(Fibonacci数列有如下特点:第1、2个数都是1,从第3个数开始,每个数都是前两个数的和)。请填空。(注:下面代码的预处理部分省略)

```
int main(){
    long f1=1,f2=1;
    int i;
    for(i=1;i<= 【1】 ;i++){
        cout<<setw(12)<<f1<<setw(12)<<f2;
        if( 【2】 ==0) cout<<endl;
        f1=f1+f2;
        f2= 【3】 ;//注意前后编程风格
    }
    return 0;
}
```

答案: **20;i%2;f1+f2**

解析:对于第三个空,由该数列的性质可以很容易答出,但要注意提示:注意前后编程风格。对于前两个空,应当清楚编写程序的依据都来源于题目要求,应对这类题型时,一定要仔细审题,把握题目所给的每个条件,同时结合代码相关信息理解题意。从题目可以看出相关信息: 1、求该数列前40个数; 2、每4列一行输出(即每输出4个数据换行一次)。观察代码可以看出:在for循环语句中,每个循环输出2个数。由此结合题意可知:输出40个数需要20次循环,4列一行需要每2个循环输出一次换行(endl或'\n')。对于for循环语句中,变量i起到计数作用,观察i的变化规律便可得出答案。

第96题

对for(表达式1;表达式3)可以理解为 **【C】**。

- A for(表达式1;表达式1;表达式3)
- B for(表达式1;表达式3;表达式3)
- C for(表达式1;1;表达式3)
- D for(表达式1;0;表达式3)

解析：循环控制语句为空，故永远为真，相当于1

第97题

程序段如下,则正确的执行结果是【C】。

```
int k=0;
while(k++<=2); cout<<k;
```

- A. 无结果
- B. 3
- C. 4
- D. 2

解析：while的条件为后置自增，故先执行表达式，再使k=k+1，当k=2时，满足条件，k=3，此时不满足条件了，由于k要自增，所以最终k=4

第98题

若i,j已定义为int型,则以下程序段中循环体总的执行次数是【 20 】(假定在循环体内不改变i,j的值)。

```
for(i=5; i; i--)
for(j=0; j<4; j++)
{ ..... }
```

解析：此语句为循环的嵌套，外循环初值i=5，执行五次外循环循环体到i=0后退出，外循环每执行一次，内循环由j=0到j=4执行四次后执行结束，故循环体的总的执行次数为5*4=20次

第99题

在do-while循环中,循环由do开始,用while 结束。必须注意的是,在while表达式后面的【分号】(用文字)不能丢,它表示do-while循环的结束。

一找出以下程序段错误(标明第几行有错，错在哪里)

(提示，从(5)行开始均有错，请给出错误原因)

- (1)const int N=10;
- (2)int s[N];
- (3)double a[20];
- (4)int n=15;
- (5)int b[n];
- (6)int c[];
- (7)s[N]={1,2,3,4,5};
- (8)for(int i=1;i<=N;i++)
- (9) cin>>s[N];

答案与解析：

应该在int 前加const，

括号中不能填n,不能整体访问

数组没有定义长度

数组不能整体访问

i应该从0开始，i<N

不能整体访问数组，应改为cin>>s[i]

第100题

以下是评分统计程序，共有8个评委打分，统计时去掉一个最高分和一个最低分，其余6个分数的平均分即是最后得分。

(提示，很多同学漏掉了某变量的初值)

```
void main()
{ float a[8],aver,max,min,sum;
  int i;
  for(i=0;i<8;i++)
    cin>>a[i];
    【1】 ;
  for(i=1;i<8;i++)
  { if(max<a[i]) max=a[i];

  if( 【2】 ) min=a[i];
  sum=sum+a[i];
}
aver=(sum-max-min)/6;
cout<<"Average="<<aver<<endl;
}
```

答案： 【1】 max=a[0],min=a[0],sum=0;

【2】 min>a[i]

解析：程序先从键盘输入初值，必须通过循环，逐个访问各个元素。然后给max,min 赋初值a[0],然后与以后的a[i]进行比大小，大的赋给max,小的赋给min。因为max,min初值是a[0],所以之后循环应该从1开始。

第101题

下列程序段执行后s的值为 (D) 。

```
int i=5, s=0;
while(i-->0)
{ if (i%2) continue;
  else s+=i;
}
```

A15 B10 C9 D6

解析：表达式i--的值为5、3、1时，进入循环内i的值为4、2、0，执行s+=i
表达式i--的值为4、2时，进入循环内i的值为3、1，是奇数，执行continue

第102题

求数列

$$sum = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{(-1)^{n+1}}{n}$$

计算该序列的前n项和，n由键盘输入。

样例输入：2

样例输出：0.5

include <iostream>

include <cmath>

using namespace std;

```

int main ()
{
    int i;
    float a,n,sum=0;
    cin>>n;
    a=1;
    for(i=1;i<=n;i++)
    {if (i%2==0)
    sum=sum-1.0/a;
    else
    sum=sum+1.0/a;
    a=a+1;
    }
    cout<<sum;
}

```

第103题

1元5角钱人民币兑换5分、2分和1分的硬币（每一种都要有）共100枚，会有很多种兑换方案。请编写程序给出各种兑换方案。

输入格式:

输入为一个正整数n，表示要求输出前n种可能的方案。方案的顺序，是按照5分硬币从少到多排列的。

输出格式:

显示前n种方案中5分、2分、1分硬币各多少枚。每行显示一种方案，数字之间空一格，最后一个数字后没有空格。

注意：如果全部方案不到n种，就顺序输出全部可能的方案。

```

#include<iostream>
using namespace std;
int main()
{
    int x,y,z,i=1,n;
    cin>>n;
    for(x=1;x<30;x++)
    for(y=1;y<75;y++)
    for(z=1;z<100;z++)
    {
        if(x+y+z==100&&5*x+2*y+z==150)
        while(i<=n)
        {
            cout<<x<<' '<<y<<' '<<z<<endl;
            i++;
            break;
        }
    }
}

```

```
return 0;
```

```
}
```

第104题

对于给定的正整数N，求它的位数及其各位数字之和。

输入格式：输入在一行中给出一个不超过 10^9 的正整数N。

输出格式：在一行中输出N的位数及其各位数字之和，中间用一个空格隔开

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
long x,n,t,s=0,c=0;
```

```
cin>>x;
```

```
while(x)
```

```
{
```

```
    t=x%10;
```

```
    x=x/10;
```

```
    s=s+t;
```

```
    c++;
```

```
}
```

```
cout<<c<<' '<<s<<endl;
```

```
return 0;
```

```
}
```

第105题

```
i=1; while(i--) { cout<<(i++); }
```

在这个程序段里，循环执行了多少次？

答案:无数次。

解析：i++, i--在判断和输出的时候都是一，所以输出无穷多个一，执行无限多次。

第106题

请写出以下程序段的输出结果

```
for(int i=0;i<5;i++)
```

```
switch(i%4)
```

```
{
```

```
case 0:
```

```
    case 1: cout<<(char)('A'+i)<<"\t";
```

```
        break;
```

```
    case 2: cout<<(char)('B'+i)<<"\t";
```

```
        break;
```

```
}
```

答案：A B D E 。

解析：当case X:后“什么都没有”而X与switch的()中的符号匹配时，就顺序执行后面的各case X:语句后的语句(不再检查后面的X是否匹配)，直到遇到break;或switch的最后一个。所以i=0的时候输出A，i=1的时候输出B，i=2时输出D，i=3时不输出但只是跳出了switch循环，还在for循环内，此时i=4，i%4=0，故输出E。

第107题

*求100之内的素数，使五个成一行。

程序分析：质数（prime number）又称素数，有无限个。

```
#include<iostream>
#include<cmath>//也可以使用C语言的旧形式，include<math.h>
using namespace std;
int main()
{
    int i,j,k,n=0;
    for(i=2;i<=100;i++){
        k=sqrt(i);
        for(j=2;j<=k;j++)
            if(i%j==0) break;
        if(j>k)
        {
            cout<<" "<<i;
            n++;
            if(n%5==0) //使其五个成一行
                cout<<endl;
        }
    }
    return 0;
}
```

【素数这道题特别经典也是常考题，做此题首先得明白素数的数学意义（一个大于1的自然数，除了1和它本身外，不能被其他自然数整除），然后在循环变量中，尽量思考怎样减少循环次数，其实此题答案的循环次数还可以减少，大家有空可以多多思考。】

第108题

解数学灯谜。有以下算式： $ABCD - CDC = ABC$ ，其中A,B,C,D为非负个位数，求ABCD的值。

```
#include<iostream>
using namespace std;
int main(){
    int A,B,C,D;
    for(A=1;A<=9;A++){
        for(B=0;B<=9;B++){
            for(C=1;C<=9;C++){
                for(D=0;D<=9;D++){
                    if((A*1000+B*100+C*10+D)-(C*100+D*10+C)==(A*100+B*10+C))
                        //注意数学表达式要用编程语言表达
                        cout<<A<<B<<C<<D<<endl;
                }
            }
        }
    }
    return 0;
}
```

分析：题目思路本质上类似“百元百鸡”或“百马百担”问题，即遍历所有可能的值并进行条件

判断，找到答案即输出。需要注意A和C作为最高位，不可能取0。基于一定数学背景，本题同样可以通过适当数学推理缩小结果的可能范围，从而对程序进行优化：很容易直接看出四位数ABCD是两个三位数的和这一事实，则必有A=1；数理基础稳固乃至雄厚者甚至可以直接推导出 $CDC-D=9*ABC$ 由此直接解出答案(滑稽)。但毕竟程序运行效率往往高于手动解题，建议适当权衡优化，避免本末倒置。

第109题

下列程序的运行结果是 357

```
int ii,x[3][3]={1,2,3,4,5,6,7,8,9};
```

```
for(ii=0;ii<3,ii++)
```

```
cout<<x[ii][2-ii];
```

解析：ii=0<3,故输出x[0][2]=3 ii=1 再输出x[1][1]=5 以此类推当输出x[2][0]=7后，ii=3，循环结束，故运行结果是357

第110题

以下程序段运行后，s的值是多少？

```
int a[]={ 1,2,3,4},i,j,s=0; //当方括号内 ( ) 为空，C++将计算元素个数
```

```
j=1;
```

```
for( i=3;i>=0;i--)
```

```
{ s=s+a[i]*j;
```

```
j=j*10;
```

```
}
```

答案：s=1234

解析：i初值为3 满足循环条件，故s=0+4*1=4

J=10 i=2 故继续执行循环体 s=4+3*10=34 i=1 依旧满足循环条件，则s=34+100*2=234 i=i-1=0 s=234+1*1000=1234 i=-1 循环结束

第五章 数组

第111题

下面说法中正确的是

A.字符个数多的字符串比字符个数少的字符串大

B.字符串“That”小于字符串“The”

C.两个字符串所包含的字符个数相同时,才能比较字符串

D.字符串“STOP□”与“STOP”相等(□表示空格)

答案：B

解析：比较字符串并不在乎两个字符串的长度，而是逐个比较两个字符串中的字符的ASCII值，若相等则继续比较，否则返回两个回字符ASCII值的差。所以a，c错误b正确。地下空格的ASCLL码大于\0的ASCLL码。

第112题

判断字符串s1是否大于字符串s2,应当使用

A.if (strcmp (s2,s1)>0)

B.if (s1>s2)

C.if (strcmp (s1,s2))

D.if (strcmp (s1,s2)>0)

答案：D

解析：判断字符串s1是否大于字符串s2，s1应该放在前面，而c项只有s1=s2时输出为假，s1不管大于s2还是小s2 if的值都为真。

第113题

下面程序的运行结果是：

```
int main()
```

```
{ char a[]="morning",t;
```

```
int i,j=0;
```

```
for(i=1;i<7;i++)
```

```
if(a[j]<a[i])
```

```
j=i; t=a[j];
```

```
a[j]=a[7];
```

```
a[7]=a[j];
```

```
puts(a);
```

```
return 0;
```

```
}
```

A.mornin

B.morning

C.mogninr

D.mo

答案：D

解析：这个题中for循环作用是将这个数组的最大值的下标赋给j，a[7]是\0，a[j](r)就变成了\0，当输出遇到\0时就停止故输出mo。

第114题

下面程序执行后的输出结果是 【 】。

```
int main()
```

```
{ char arr[2][4];
```



```

strcpy(arr[0], "you");

strcpy(arr[1], "me");

arr[0][3]= '&'; cout<<arr[0];

return 0;

}

```

A.me
B.you
C.err
D.you&me

答案：D

解析：由于a[0][3]='&';这一步把a[0][3]的'&'覆盖了，所以输出arr[0]时arr[1]也被输出来了。

第115题

请填空,并回答问题.

(提示:访问字符串时,通常用串结束标志'\0'来判定字符串是否结束。
cin.getline(参数1, 参数2);//参数1是数组名, 参数2是最多字符个数。
提问:语句1的n值多少?n是哪个元素的下标?程序功能实现了什么计数?

```

int main()
{char str[100];
int n;
cin.getline( 【1】 );//输入:abcdef
for(n=0;str[n]!='\0';n++) ;
cout<<n<<endl;//语句1
}

```

答案:

str,99,'\0'

n值是6, 是'\0'的下标, 实现了元素个数计数

第116题

若输入: Hello2018,C++100! 给出输出结果, 判断其功能, 再回答问题。

(提示:字符串中的数字为字符, 须加单引号)。

提问:1)语句1中将单引号去掉行否? 为什么? 2)若去掉语句2, 输出结果? 为什么?

```

int main()
{ char word[50];
cin>>word;
int pos=0;

```

```

for(int i=0; word[i]!='\0'; i++)
if(word[i]<'0' || word[i]>'9') //语句 1
{ word[pos]=word[i];
++pos; }
word[pos]='\0'; //语句 2
cout<<"result: "<<word<<endl;
return 0;
}

```

答案:

不行，需要比较ASCII码。

输出为result: Hello2018,C++100!没有终止符导致数组2继续输出

第117题

删除一个字符串中的所有空格，先填空，后回答问题。

(提示:注意调用函数时的实参与形参的类型个数一致)

提问:1) 若去掉语句1，输出结果是什么？为什么？

```

void DelBlank(char s[]);
void DelBlank(char s[])
{int i=0,j=0;
for(; s[i]!='\0'; i++)
{if(s[i]!=' ')
{s[j] = s[i];
【2】 ;}
}
s[j]= 【3】 ;//语句1,去掉此句，输出结果？
}
int main()
{char s[]="How are you";
DelBlank( 【4】 );
cout << s;
return 0;}

```

答案: 【1】 '\0'

【2】 j++

【3】 '\0'

【4】 s

输出结果是Howareyouou，因为没有终止符导致采用了原数组的元素

第118题

向升序字符串中插入1个字符，使得插入后的字符仍然保持有序。

提示，cin.getline(参数1，参数2);//参数1是数组名，参数2是最多字符个数。

提问:1)语句1的n值多少？n是哪个元素的下标？

2) 为何在语句2的n+1位置存'\0'？

3)语句3的for循环功能？

```

void insertStr (char str[], char c);
int main()
{ char str[100],c;
cin.getline( 【1】 );//输入abdfg
cin>>c;
insertStr( 【2】 );
cout<<str<<endl;
return 0;
}
void insertStr (char str[], char c)
{
int n,i;
for(n=0;str[n]!='\0';n++) ;
cout<<n<<endl; //语句1
str[n+1]='\0';//语句2
for(i=n-1;i>='\0';i--)//语句3
if( 【3】 )
str[i+1]=str[i];
else break;
【4】 =c;//插入字符c
}

```

答案:

str,99

str,c

str[i]>c

str[i+1]

5 终止符所在下标

终止数组

从后面往前——检测数组原元素与输入数字的大小，遇到大的往后一位储存

第119题

删除字符串中指定字符

输入格式：在第一行中输入一个长度不超过20的字符串，并以回车结束；在第二行中输入待删除的一个字符，并以回车结束。

输出格式：在一行中输出删完指定字符的字符串。

解析：此题为“删除一个字符”的升级版。若将删除字符之后的数组元素逐个前移则步骤将变得相当复杂。利用变量对数组元素定位，手动复制数组并跳过被删除元素的方法较为可行，且该方法对许多类型数组编程题具有普适性。

代码如下：

```

#include<iostream>
using namespace std;
int main(){
    char a[20],x;

```



```

int i,j;
cin.getline(a,20);
cin>>x;
for(i=j=0;s[i]!='\0';i++)
    if(a[i]!=x) a[j++]=a[i];
s[j]='\0';
for(i=0;a[i]!='\0';i++) cout<<a[i];
return 0;
}

```

第120题

从键盘输入杨辉三角的行数 n ($n \leq 15$)，输出 n 行杨辉三角，每个数据占5列。

```

#include<iostream>
#include<iomanip>
using namespace std;
int main(){
    int n,a[16][16]; //为了行数、列数同下标一致,空出0行和0列,方便阅读与修改程序
    cin>>n;
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i;j++){ //杨辉三角的第n行共有n个数
            if(j==1||j==i) a[i][j]=1; //每行的首位及末位数为1
            else a[i][j]=a[i-1][j]+a[i-1][j-1]; //其余的数为其左上和其正上两数之和
        }
    }
    for(int i=1;i<=n;i++){
        for(int j=1;j<=i;j++){
            cout<<setw(5)<<a[i][j]; //每个数据占5列,用iomanip中的setw函数
        }
        cout<<endl; //需要输出多行数据时切记换行
    }
    return 0;
}

```

第121题

如果输入ABC,则下面程序执行后的输出结果是 **【 】**。

```

int main()
{
    char ss[10]="1,2,3,4,5";
    gets(ss);
    strcat(ss, "6789");
    cout<<ss; return 0;
}

```

```
}
```

答案：ABC6789

解析：gets(s1)的功能：接收一个字符串，不以空格作为字符串输入结束的标志，而只以回车作为输入结束。输入的字符串ABC重新赋给字符串数组ss，strcat(s1,s2)的功能：把字符串2中的字符串连接到字符串1中字符串的后面，并删去字符串1后的串标志'\0'，因此输出ABC6789

第122题

不能把字符串："good!"赋给数组 b 的语句是【 】。

- A) char b[10]={ 'g', 'o', 'o', 'd', '!', ' '};
- B) char b[10];b="good!";
- C) char b[10];strcpy(b, "good! ");
- D) char b[10]= "good!";

答案：B

解析：不能用"="对字符串数组单独赋值，因为b为数组名，为地址，不能被赋值

第123题

数组作为函数参数,求平均值,请回答问题。

提问:1) 语句1,调用函数Input(s[],10);对不对?

2)语句2, 通过循环逐个访问数组a各元素, 实际访问的是哪个数组?

3)能否将语句3改为: cin>>s[i]; ?子函数有定义s么?

4)语句4, 在子函数另外定义数组b有意义么? 能接收实参传递来的数组首地址么?

5)语句5, 能否用return将数组返回?

```
void Input(float a[10],int n);
float average(float a[10],int n);
int main()
{
    float s[20];
    int n;
    cin>>n;
    Input(s[],n); //语句1
    cout<<average(s,10);
}
void Input(float a[],int n)
{
    for(int i=0;i<n;i++) //语句2
        cin>>a[i]; //语句3
}
/*以下程序段位注释, 与上面对比
void Input2(float a[],int n){
    float b[20]; //语句4
    for(int i=0;i<n;i++)
        cin>>b[i];
```

```

    return b;//语句5
}*/
float average(float a[],int n){
    int i; float sum=0;
    for(i=0;i<n;i++)
        sum=sum+a[i];
    return (sum/n);
}

```

答案： 1) 不对，实参应该是数组名s，而且n是由用户自己输入的，不恒为10。
 数组s。因为这个函数中的a是形参
 不能，没有。只有形参数组a，没有定义数组s
 无意义，不能
 5) 不能，数组不能整体返回

第124题

在C/C++语言中，数组的首地址是（数组名）

第125题

对于数组a[N]来说，使用数组的某个元素时，下标最大值是（N-1）

第126题

填空：将数组元素逆序存放，即最前面与最后面的元素交换，依次类推。

```

void main()
{const int N=10;
int a[N]={1,2,3,4,5,6,7,8,9,10},i,j,t;
for(i=0,j=N-1;i<j; 【1】 )
{ 【2】 ;//语句1
}
for(i=0;i<N;i++) //语句2
cout<<setw(3)<< 【3】 ;
}

```

答案： 【1】 i++， j--
 【2】 t=a[i];a[i]=a[j];a[j]=t;
 【3】 a[i]

解析：这道题的思路就是交换顺序，第一个数组元素和最后一个交换，第二个数组元素和倒数第二个交换。比如，a[0]与a[N-1]交换,a[1]与a[N-2]交换。交换结束后再数出来。

第127题

求数组的最小元素及其下标。在主函数中从键盘输入一个正整数n（1<n≤10）。int getmin(int a[], int size);

```

void main()
{ int n,k;
cout<<"请输入n(1<n<=10):";
cin>>n;
int a[ 【1】 ]; //定义数组时长度必为常量
for(int i=0;i< 【2】 ;i++)

```



```

    cin>>a[i];
    k= 【3】 ;//语句3, 注意调用函数时的实参
    cout << "The min is : " << 【4】 << endl;
    cout << "The index is : " <<k << endl;
}
int getmin( int a[], int size )
{ int i, index = 0;
  for( i=0; i<size; i++ )
  { if( 【5】 )
      index = i; //语句4
  }
  return  index;
}

```

答案: 【1】 10 【2】 n 【3】 getmin(a,n) 【4】 a[k] 【5】 a[index]>a[i]

解析: 定义数组时一定要用常量。调用函数时要注意实参, 【3】 填getmin(a[],10)时是错误的。

第128题

假设定义数组int a[10];则程序中使用a[10]=1;, 表示 (D)

A 将数组a的最后一个元素赋值为1

B将数组a的第一个元素赋值为1

C、将数组的所有10个元素都赋值为1

D、将数组a的最后一个元素之后的空间赋值为1, 可能导致严重后果

解析: 定义的数组不存在a[10],所以, 可能会出错

第129题

对于给定的正整数N, 求它的位数及其各位数字之和。

提问:1)语句1的const能否去掉?为什么?

2)语句2功能:逐个输入各元素, 【1】 填a[N]对不对?

3) 【2】 填0;对不对? 语句3能否去掉?为什么?

```

int main()
{ const int N=10;//语句1
  int a[N];
  int i,index,max;
  for(i=0;i<N;i++) //语句2
  cin>> 【1】 ;
  max= 【2】 ;//语句3
  for(i=0;i<N;i++)
  if(a[i]>max)
  { 【3】 ;}
  cout<<"MAX="<<max;
}

```

答案与解析: a[i]、a[0]、max=a[i]

不能, 数组的下标不能是变量, const使得N变为常量。

不对, 因为输入的数可能都小于零, 此时程序有误。

可以, 对, max初值对最后结果没有影响。语句三的作用是用max储存每次循环判断出的最

大值。

第130题

求数组的最小元素及其下标。在主函数中从键盘输入一个正整数 n ($1 < n \leq 10$)。

(提示: 【1】 填 n 不对)

【3】 填: `getmin(a[], 10)`不对,

【5】 填: `a[i] < index`不对)

提问: 语句4的`index`存的元素值还是元素下标?

```
int getmin( int a[], int size );
```

```
int main()
```

```
{ int n, k;
```

```
cout << "请输入n(1<n<=10).:";
```

```
cin >> n;
```

```
int a[ 【1】 ]; //定义数组时长度必为常量
```

```
for(int i=0; i< 【2】 ; i++) //语句2
```

```
cin >> a[i];
```

```
k= 【3】 ; //语句3, 注意调用函数时的实参
```

```
cout << "The min is : " << 【4】 << endl;
```

```
cout << "The index is : " << k << endl;
```

```
}
```

```
int getmin( int a[], int size )
```

```
{ int i, index = 0;
```

```
for( i=0; i<size; i++ )
```

```
{ if( 【5】 )
```

```
index = i; //语句4
```

```
}
```

```
return index;
```

```
}
```

答案与解析:

【1】 100 参考值, 随便填

【2】 n

【3】 `getmin( a, n)`

【4】 `a[k]`

【5】 `a[i] < a[index]`

不对, 定义数组时长度必为常量。可以写100

不对, 注意调用函数时的实参

不对, `index`是下标, 需要比较的是元素

第131题

编写程序, 利用折半查找法从一个升序排列的数列中查找某数是否存在, 如果存在则输出该数的下标, 否则输出NO。输入分3行: 第一行为 n 的值, 代表数列中数的个数, 第二行为 n 个数(升序排列), 第三行为要查找的数。

样例输入:

5

0 1 2 3 5

3

样例输出: 3

样例输入:

5

0 1 2 3 5

4

样例输出: NO

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int a[100]={0},b[100]={0},i,n,c,d=1,e,f,s,t,x;
```

```
cin>>n;
```

```
for(i=1;i<=n;i++)
```

```
cin>>a[i];
```

```
cin>>x;
```

```
for(i=1;i<=n;i++)
```

```
{
```

```
if(a[i]==x)
```

```
break;
```

```
else
```

```
d++;
```

```
}
```

```
if(d<n+1)
```

```
cout<<d-1;
```

```
else
```

```
cout<<"no";
```

```
}
```

第132题

编写程序，将一个数组中的数循环左移。输入分2行：第一行为n的值，第二行为n个数。

样例输入:

5

1 5 3 2 0

样例输出: 5 3 2 0 1

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{int a[100]={0},b[100]={0},i,n,x;
```

```
cin>>n;
```

```
for(i=1;i<=n;i++)
```

```
cin>>a[i];
```

```
for(i=1;i<n;i++)
```

```
b[i]=a[i+1];
```

```
b[n]=a[1];
```



```
for(i=1;i<=n;i++)
cout<<b[i]<<" ";
}
```

第133题

若二维数组a有m列,则在a[i][j]前的元素个数为 **【B】**。

- A. $j*m+i$
- B. $i*m+j$
- C. $i*m+j-1$
- D. $i*m+j+1$

解析：由题得二维数组a有m列，故该数组中的元素可以写成：

A00 A01 A02——A0(m-1)

A10 A11 A12——A1(m-1)

则可以认为i代表行数 j代表列数，则a[i][j]前的元素数为 $i*(m+1)+j=i*m+j$

故选择B

第134题

执行下面的程序段后,变量k中的值为 **【B】**。

```
int k = 3, s[2];
```

```
s[0] = k;
```

```
k = s[1] * 10;
```

- A. 30
- B. 不定值
- C. 33
- D. 40

解析：int s[2]这样定义的数组没有初始化，s[1]的值是不确定的k=s[1]*10也是不定值了

第135题

若有说明:int a[3][4];则对a数组元素的非法引用是 **【A】**。

- A. a[0][4]
- B. a[4-2][0]
- C. a[1][3]
- D. a[0][2*1]

解析：A选项非法，注意数组元素的下标从0开始，故有四个元素时，数组的下标只能是0, 1, 2, 3, 其它选项均合法

第136题

有说明语句:int a[10]={6,7,8,9,10}; 则下面说法中正确的是 **【B】**。

- A. 将5个初值依次赋给a[6]至a[10]
- B. 将5个初值依次赋给a[0]至a[4]
- C. 将5个初值依次赋给a[1]至a[5]
- D. 因为数组长度与初值的个数不相同,所以此语句不正确

解析：数组可以按一维形式进行赋值，将所给初值依次赋给a[0]——a[4]，初值列表提供的元素个数不能超过数组长度，但可以小于数组长度，如果小于数组长度,则只初始化前面的数组元素，剩余元素初始化为0

第137题

若有语句int a[8],则下列对a的描述中正确的是【D】。

- A. 说明a[8]是整型变量
- B. 定义了一个数组a,数组a共有9个元素
- C. 其它三个答案均不正确
- D. 定义了一个名称为a的一维整型数组,共有8个元素

解析: int a[8] 定义了一个名称为a的一维整型数组,共有8个元素, D选项正确

第138题

数组逆序

两种思路:

思路一: 借助另一个数组实现“逆序”;

```
#include<iostream>
#include<iomanip>
using namespace std;
int main(){
    int n,a[100],b[100],i;
    cin>>n;
    for(i=0;i<n;i++){
        cin>>a[i];
        b[(n-1)-i]=a[i];
    }
    for(i=0;i<n;i++){
        cout<<setw(4)<<b[i];
    }
    return 0;
}
```

思路二: 直接逆序输出原数组

```
#include<iostream>
#include<iomanip>
using namespace std;
int main(){
    int n,a[100],i;
    cin>>n;
    for(i=0;i<n;i++){
        cin>>a[i];
    }
    for(i=n-1;i>=0;i--){
        cout<<setw(4)<<a[i];
    }
    return 0;
}
```

第139题

编写函数,删除数组中指定位置处的元素,删除数组中的某个元素时,其后的所有元素需要顺序向前移动。

裁判测试程序样例：

```
#include<iostream>
using namespace std;
void delNum(int a[],int n,int x);
int main()
{
    const int N=10;
    int a[N]={10,111,102,213,124,415,176,170,179,19},x;
    cout<<"请输入待删除元素的位置(0~9):";
    cin>>x;
    delNum(a,N,x);
    for(int i=0;i<N-1;i++)
        cout<<a[i]<<" ";
    return 0;
}
/* 你的代码将被嵌在这里 */
```

两种思路：

思路一：

```
void delNum(int a[],int n,int x){
    for(int i=x;i<n-1;i++){
        a[i]=a[i+1];
    }
}
```

思路二：

```
void delNum(int a[],int n,int x){
    int b[n];
    for(int i=0,j=0;j<n;i++,j++){
        if(j==x) j++; //直接跳过a[x]这一元素
        b[i]=a[j]
    }
    for(int i=0;i<n-1;i++){
        a[i]=b[i];
    }
}
```

解析：本题只要求删除一个元素，思路一的方法更为简单直接，容易想到；思路二巧妙在定义另一个数组，借助变量*i,j*分别在两个数组中定位并传递元素的值。这种方法往往对于操作略复杂的数组题较为常用，例如删除多个数组元素等。

第140题

键盘输入*n*(*n*≤20)个数存入一维数组中，编程实现将其中的值前后倒置后重新存入该数组中并输出。

输入格式：

第一行输入数据个数*n*，第二行输入*n*个数据，数据之间用空格隔开。

输出格式：

输出数据之间用逗号分隔，最后换行。

输入样例：

10

1 2 3 4 5 6 7 8 9 100

输出样例:

100,9,8,7,6,5,4,3,2,1

```
#include <iostream>
using namespace std;
int main()
{
    int n,i,t,a[20];
    0<n<=20;
    cin>>n;
    for(i=0;i<n;i++)
    cin>>a[i];
    for(i=0;i<n/2;i++)
    {
        t=a[i];
        a[i]=a[n-i-1];
        a[n-i-1]=t;
    }
    for(i=0;i<n;i++)
        if(i==0)
            cout<<a[i];
        else
            cout<<","<<a[i];
    return 0;
}
```

第141 题

输入6个整型数据（6个数不可以相同；正负均可），用空格隔开，并放入数组中（如：int Array[6];），求去掉最大值和最小值后剩下的4个数的和。

输入格式:

输入6个整型数据（6个数不可以相同；正负均可），用空格隔开。

输出格式:

输出去掉最大值和最小值后剩下的4个数的和。

输入样例:

1 2 3 4 5 6

输出样例:

14

```

#include <iostream>
using namespace std;
int main()
{
    int i,a[6],min,max,sum=0,s;
    for(i=0;i<6;i++) cin>>a[i];
    min=a[0];
    max=a[0];
    for(i=0;i<6;i++)
    {
        if(a[i]>max) max=a[i];
        if(a[i]<min) min=a[i];
    }
    for(i=0;i<6;i++)
    {
        sum=sum+a[i];
    }
    s=sum-max-min;
    cout<<s;
}

```

解题思路：先输入数组元素，然后每个元素与max或者min比较找出max与min，然后算出所有元素之和再减去max、min。

第142题

将一个具有20个元素的数组中的中间10个元素按从大到小顺序排序，要求使用冒泡法排序。

输入格式:

在一行中输入20个小于50的整数，数据之间只能用1个空格间隔。

输出格式:

直接输出变化后的数组，每个数输出占4列列宽。

输入样例:

5 4 3 2 1 8 5 2 9 6 3 1 4 7 4 1 2 3 4 5

输出样例:

```

5 4 3 2 1 9 8 7 6 5 4 4 3 2 1
1 2 3 4 5

```

```

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

```

```

int main()
{
    int a[20],i,j,t;
    for(i=0;i<20;i++) cin>>a[i];
    for(i=5;i<=14;i++)
    for(j=5;j<=18-i;j++)
    {
        if(a[j]<a[j+1])
        {
            t=a[j];
            a[j]=a[j+1];
            a[j+1]=t;
        }
    }
    for(i=0;i<20;i++) cout<<setw(4)<<a[i];
    return 0;
}

```

总结：基本思路：每次将相邻的两个数比较，将较大数调到前面。重复这个步骤，最终将最大数排在最后。

算法描述：

- 1.比较相邻两个数，若前面的大于后面的则交换，否则不交换（保证大数在后面）；
- 2.对每一个相邻的数做同样1的工作，这样从开始相邻一对，到结尾相邻一对，最后的数就是最大的数。
- 3.余下的元素进行上面的操作(最后那个最大数不再参与比较)。
- 4.重复1~3步骤，直到顺序完成。

比如，对下面这个序列进行从小到大排序(90 21 132 -58 34)

第0轮比较($i=0$, i 代表轮数): 90 21 132 -58 34(原顺序)

第0次($j=0$): 90和21比, ($a[0]>a[1]$)?则互换位置: 21 90 132 -58 34

第1次($j=1$): 90和132比, ($a[1]>a[2]$)?则不用交换位置: 21 90 132 -58 34

第2次($j=2$): 132和-58比, ($a[2]>a[3]$)?则互换位置: 21 90 -58 132 34

第3次($j=3$): 132和34比, ($a[3]>a[4]$)?则互换位置: 21 90 -58 34 132

(第 j 次: $a[j]$ 与 $a[j+1]$ 比较)

到此第0轮结束,共比较4次($j=0\sim3$ 次, 即 $0\sim N-2$ 次), 第0轮的结果是找到了最大数, 并浮到了最右边。

for($j=0$; $[1]$; $j++$) // j 代表每轮的次数

if($a[j]>a[j+1]$) //第 j 次: $a[j]$ 与 $a[j+1]$ 比, 若前面的大于后面的则交换, 否则不交换
{交换 $a[j]$ 与 $a[j+1]$ }

第143题

本题要求实现一个对数组进行循环左移的简单函数：一个数组 a 中存有 n (>0) 个整数，在不允许使用另外数组的前提下，将每个整数循环向左移 m (≥ 0) 个位置（最前面的 m 个数循环移至最后面的 m 个位置）。如果还需要考虑程序移动数据的次数尽量少，要如何设计移动的方法？

输入格式:

输入第1行给出正整数 n (≤ 100) 和整数 m (≥ 0) ; 第2行给出 n 个整数, 其间以空格分隔。

输出格式:

在一行中输出循环左移 m 位以后的整数序列, 之间用空格分隔, 序列结尾不能有多余空格。

输入样例:

8 3

1 2 3 4 5 6 7 8

输出样例:

4 5 6 7 8 1 2 3

```
#include <iostream>
using namespace std;
int main()
{
    int a[100],n,m,i,j,temp;
    cin>>n>>m;
    for(i=0;i<n;i++) cin>>a[i];
    m=m%n;
    for(i=0;i<m;i++)
    {
        int temp;
        temp=a[0];
        for(j=0;j<n-1;j++) a[j]=a[j+1];
        a[n-1]=temp;
    }
    for(i=0;i<n;i++)
    {
        if(i==0) cout<<a[i];
        else cout<<" "<<a[i];
    }
}
```

第144题

二维数组应用, 写出程序的输出结果,并回答问题。

提问:

- 1) 语句1中的 i 作用?
- 2) 语句2中的 j 作用?
- 3) 语句3挪至语句4所在位置行否?为什么?
- 4) 语句4的那对 $\{\}$ 去掉, 会怎样?

```
#include<iostream>
#include<iomanip>
using namespace std;
void main()
{ int i,j,k,m;
  int array1[3][2]={4,2,5,6};
```

```
int array2[3][2]={{4,2},{5},{6}};
for (i=0;i<3;i++) //语句1
{ for (j=0;j<2;j++) //语句2
{cout <<setw(2) <<array1[i][j];}
cout <<endl; //语句3
} //语句4
for ( k=0;k<3;k++)
{ for ( m=0;m<2;m++)
cout <<setw(2) <<array2[k][m];
cout <<endl;
}
}
```

答案：循环行

循环列

不行，会导致输出时做不到每行换行

不行，会导致只换一行

第145题

按以下形式输出杨辉三角形，请填空。

(提示:数组0行未使用,0列未使用)

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

.....

```
void main()
{int i,j,a[10][10];
for(i=1,【1】;i<10;i++,j++)
{a[i][1]=1;//1列元素
a[i][j]=1;//对角线元素
}
for(【2】;i<10;i++)
for(j=2;【3】;j++)
a[i][j]=【4】;
for(i=1;i<10;i++)
{ for(j=1;【5】;j++)
{cout<<setw(4)<<a[i][j];}
cout<<endl;}}
```

答案：1) j=1

2) i=3

3) j<i-1

4) a[i-1][j-1]+a[i-1][j]

5) j<i

第146题

找出二维数组a中每行的最大值，并按一一对应的顺序放入一维数组s中。即：第零行中最大值放入s[0]中，第一行中最大值放入s[1]中.....，然后输出每行的行号和最大值。请填空。

提问:1) 语句1的功能? 【1】 填0行否?

```
void main()
{int a[4][4],s[4],i,j,k;
for(i=0;i<4;i++)
for(j=0;j<4;j++)
cin>>a[i][j];
for(i=0;i<4;i++)
{s[i]= 【1】 ;//语句1
for(j=1;j<4;j++)
if(s[i]<a[i][j]) 【2】
}
for(i=0;i<4;i++)
cout<<"Row="<<i<<"Max="<<s[i]<<endl;
}
```

答案:

a[i][0]

s[i]=a[i][j]

不行，可能都是负值

第六章 附录

C++知识点部分总结:

标识符: 只能由字母和下划线还有数字组成; 第一个字母必须是字母或下划线, 不能是数字;

指数形式: 4E4表示 4×10^4 【E和e都可以】

字符常量和字符串常量: 'a' "a"

类型转换的优先级: char<int<float<double;

=是赋值, ==是等于