C++重点知识点总结及习题

第 1 章 C++的特点

C++的特点： 1.支持抽象数据类型

2.多态性，一个接口，多重算法，支持早期联编和滞后联编

3.继承性，保证了代码复用，支持分类的概念

一个 c++程序由一个或多个函数构成，并且在这些函数中只有一个主函数 main，它是程序执行的入口。

C++程序严格区别字母的大小写。

第 2 章 基本数据类型、运算符与表达式

【典型例题】

例题 1：对以下各种数据类型比较所占用存储空间的大小：

(1) char、int、short int、long int、double、long double.

(2) signed int、unsigned int.

解答：

(1) 本题主要考查的知识点是各种类型所占用的存储空间的大小以及相应的表数范围。在 32 位计算机中，char 占 1 个字节，short int 占 2 个字节，int 占 4 个字节，long int 占 4 个字节，double 占 8 个字节。sizeof 运算符用于求解某种数据类型的大小。short 和 long 在修饰整型时可以省略 int。答案为：

$$1 \equiv sizeof(char) \leq sizeof(short) \leq sizeof(int) \leq sizeof(long) \leq sizeof(double)$$

(2)对于一种确定的数据类型有符号数与无符号数所占的存储空间相同，表数范围不同。修饰符 signed 和 unsigned 只能用于修饰字符型和整型。答案为：

例题 2：下列哪一项能用作用户自定义的标识符：

(a)const (b)2var (c)my name (d)var2

解答：

本题主要考查标识符命名规则。C++关键字不能用于用户自定义标识符，(a)中 const 是关键字；第一个字符必须是字母或下划线，(b)中 2var 是以数字开头；不能含有空格，(c)中 my name 含有空格。答案为：d。

例题 3：指出下列程序中的错误：_____。

```
int main()
{
    const int x;
    x=100;
    return 0;
    }
```

解答：

本题主要考查对符号常量的理解。const 定义的符号常量必须初始化,由 const 定义的常量的值不可以改变。所以本题有两处错误：第一、没有对符号常量 x 进行初始化；第二、给符号常量赋值是错误的。

例题 4：给下列表达式加上全部的括号(假设所用变量均已定义)：

(1) a+b-c--%b

(2) a>b?b:c>d?a:c<a?c:d

(3) a+=a+b||e

(4) a&b+c++

(5) -a&&b-c

(6) k=b=c=a

解答：

本题主要考查表达式中运算符的优先级与结合性。请参阅表 2.1。为了避免出错，建议读者在书写表达式时完整书写括号。

(1)答案为：(a+b)-((c--)%b)
(2)答案为：a>b?b:(c>d?a:(c<a?c:d))
(3)答案为：a+=((a+b)||e)
(4)答案为：a&(b+(c++))
(5)答案为：(-a)&&(b-c)
(6)答案为：k=(b=(c=a))

例题 5：请根据下列题意写出相应的表达式。

(1) 有 a、b、c、max 四个变量 a、b、c 中的最大值，并将结果放入 max 中。
(2) 年龄在 1 到 100 之间（包含 1 和 100，年龄用变量 age 表示）。
(3) 公式$\frac{1}{2}(a+b)h$。
(4) 判断一年是否为闰年，年用 year 表示。满足下列两个条件之一即为闰年：①能被 4 整除但不能被 100 整除 ②能被 400 整除。

解答：

（1） 主要考查对条件表达式的理解和书写。答案为：max=a>b?(a>c?a:c):(b>c?b:c)。
（2） 主要考查对逻辑表达式的理解和书写。答案为：1<=age&&age<=100。
（3） 主要考查如何在计算机中表示一个数学公式。答案为：(a+b)*h/2。
（4） 主要考查对逻辑表达式的理解和书写。答案为：(year%4==0&&year%100!=0)||(year%400==0)。

例题 6:下列选项中两个表达式的运算结果相同的是（ ）。

(a)3/2 和 3.0/2.0 (b)3/2 和 3.0/2 (c) 3/2.0 和 3.0/2.0 (d) 3/2.0 和 3/2

解答：

本题考查数据类型及表达式中数据类型的隐式转换。3/2 中两个操作数都为整型，运算结果仍为整型即 1；3.0/2 和 3/2.0 中一个操作数为整型另一个为浮点型，运算时整型隐式转换为浮点型，运算结果也为浮点型即 1.5；3.0/2.0 两个操作数均为浮点型，结果也为浮点型即 1.5。答案为：(c)。

例题 7:

下列程序的运行结果为：_____。

```
#include <iostream>
void main()
{
    int a=2,b=4,i=0,x;
    x=a>b&&++i;
    cout<<"x："<<x<<endl;
    cout<<"i："<<i<<endl;
}
```

解答：

本题主要考查“短路”表达式的运算。对于表达式中的“与”运算而言，只要有一个操作数为假，结果为假。所以当第一个操作数为假时，不在求解其它操作数。对于表达式中的“或”运算而言，只要有一个操作数为真，则结果为真。所以当第一个操作数为真时，不在求解其它操作数。本题中 a>b 为假，所以表达式 a>b&&++i 为假，而++i 没有执行。故 i 为 0。答案为:x:0
i:0

例题 8:求解下列各表达式的值（其中 x 的值为 100）。

(1) (a=1,b=2,c=3)
(2) 1|3<<5
(3) 'a'+3&&!0%1

(4) x%2? “odd”:” even”

解答：

(1) 逗号表达式的值是其最后一个表达式的值。答案为：3。

(2) <<运算符的优先级高于|运算符，所以先算 3<<5 结果为 96（二进制 1100000），然后与 1 做按位与运算则结果为 97（二进制 1100001）。答案为：97。

(3) 参与本题的运算符，按优先级由高到低依次是：!运算符、算术运算符、逻辑运算符。'a'+3 时字符型首先隐式转换成整型然后相加结果为 100，!0%1 即 1%1 结果为 0，100&&0 结果为 0。答案为：0。

(4) 算术表达式的优先级高于条件表达式，所以先算 x%2 结果为 0，0?”odd”:”even”结果为”even”。本题完成判断一个数是奇数还是偶数，若该数为奇数，则表达式的值为”odd”，为偶数，则表达式的值为”even”。答案为：”odd”。

例题 9:下列程序运行结果为：_____。

```
#include <iostream>
#include <iomanip>
void main()
{
    int a=23;
    double b=23.123456789;
    cout<<a<<'\t'<<b<<endl;
    cout<<setprecision(0)<<b<<endl;
    cout<<setiosflags(ios::fixed)<<setprecision(7)<<b<<endl;
    cout<<setiosflags(ios::scientific)<<b<<endl;
    cout<<setprecision(6);
    cout<<setiosflags(ios::showbase);
    cout<<hex<<a<<'\t'<<a<<endl;
    cout<<dec;
    cout<<setw(10)<<setfill('*')<<setiosflags(ios::left)<<a<<endl;
    cout<<setfill(' ');
}
```

解答：

本题主要考查对格式化输入输出的掌握。

①本题主函数中第三行输出 a,b,'\t'为转义字符，其含义是跳过一个制表位。不设置输出宽度时，默认输出 6 位有效数字，超出部分四舍五入。所以该行输出为：23　　23.1235。

② setprecision(n)设置显示精度，最少显示一位有效数字。如果不重新设置，则其保持效力，所以使用完后要还原为 6 位默认值。第四行中设置 setprecision(0)与 setprecision(1)作用相同，结果显示一位有效数字即为：2e+001。

③ setiosflags(ios::fixed)为固定的浮点显示，其后跟 setprecision(n)表示小数点后显示精度为 n。所以第五行输出结果为：23.1234568。

④ setiosflags(ios::scientific)为指数显示，当其整数部分宽度大于设置的显示精度(默认为 6 位）时，以指数形式显示结果。否则根据设置的（或默认的）显示精度显示 n 位有效数字。所以第六行输出结果为：23.12346。

⑤ setiosflags(ios::showbase)为指定在数值前输出进制。hex 置基数为 16，且该操作保持效力，所以使用完后应该恢复为默认值 10 进制。第九行输出结果为：0x17　　0x17。

⑥setw(n) 设域宽为 n 个字符，setfill(c) 设填充字符为 c ，setiosflags(ios::left)为左对齐。第十一行输出结果为：23********。

答案为：

```
23      23.1235
2e+001
23.1234568
23.12346
```

0x17　　0x17
23********

【习题】

一、　选择题

1. 下列数据类型不是 C++语言基本数据类型的是（　）。
   (a）字符型　　(b）整型　　(c）浮点型　　(d）数组

一、下列字符列中，可作为 C++语言程序自定义标识符是（　）。选择题

2.
   (a)x　　(b）-var　　(c)new　　(d)3i

3. 下列数中哪一个是 8 进制数（　）。
   (a)0x1g　　(b)010　　(c)080　　(d)01b

4. 已知 a=1,b=2,c=3，则表达是++a||-b&&++c 的值为（　）。
   (a)0　　(b)1　　(c)2　　(d)3

5. 下列表达式选项中，（　）是正确的。
   (a)++(a++)　　(b)a++b　　(c)a+++b　　(d)a++++b

6. 已知枚举类型定义语句为：（ ）。
   enum color{RED,BLUE,PINK=6,YELLOW,GREEN,PURPLE=15};
   则下列叙述中错误的是（　）。
   (a)枚举常量 RED 的值为 1　　(b)枚举常量 BLUE 的值为 1
   (c)枚举常量 YELLOW 的值为 7　　(d)枚举常量 PURPLE 的值为 15

7. 下列程序的运行结果正确的是（　）。

```
#include <iostream>
#include <iomanip>
void main()
{
    const double pi=3.1415926;
    cout<<setprecision(3)<<pi<<endl
        <<setiosflags(ios::fixed)<<pi<<endl
        <<setprecision(8)<<setfill('*')<<setw(12)<<pi<<endl;
    return;
}
```

   (a) 3.142
   3.142
   **3.14159260
   (b) 3.14
   3.142
   **3.14159260
   (c) 3.14
   3.14
   3.14159260**
   (d) 3.14
   3.142
   ***3.1415926

8. 若 int x=3,y=5;则表达式 x&y++%3 的值为（　）。
   (a)0　　(b)1　　(c)2　　(d)3

9. 下列常量正确的是（　）。

(a) ”hello World”　(b)1FL　(c) 3.14UL　(d) 1.8E-3

10. 若 char x=97;,则变量 x 包含几个字符（　）。

(a)1 个　(b)2 个　(c)4 个　(d)8 个

二、 填空题

1. c++语言标示符是以字母或_____开头的，由字母、数字、下划线组成。
2. 在 C++语言中，char 型数据在内存中的存储形式是_____。
3. 在内存中，存储字符‘x’占用 1 个字节，存储字符串“x”要占用_____个字节。
4. 符号常量可以用宏定义 define 和_____表示。
5. 转义字符序列中的首字符是_____。
6. 空字符串的长度是_____。
7. 表达式 cout<<’\n’;还可以表示为_____。
8. 若要为 unsigned int 定义一个新的名字 UINT 应采用的语句是_____。
9. 以下程序不借助第 3 个变量实现 a,b 值得交换，请填空。

```
#include <iostream>
Void main()
{
int a,b;
cout<<”输入 a,b:”;
cin>>a>>b;
a=___a+b_____;
b=_____a-b_____;
a=_____(a-b)/2;
cout<<a<<’\t’<<b<<endl;
}
```

10. 大多数 C++程序都要包含头文件_____。该文件中包含了所有输入/输出流操作所需的基本信息。当使用带参数的操作时，程序中必须包含头文件_____。

三、 编程题

1. 编写一个程序，输入一个三位数，分别输出该数的百位、十位和个位。
2. 编写一个程序打印出各种基本数据类型、几个你自己定义的复合数据类型的大小。使用 sizeof 运算符。

【参考答案】

一、选择题

1-5） dabbc 6-10）abcda

二、填空题

1. 下划线
2. ASCII
3. 2
4. const
5. \
6. 0
7. cout<<endl;
8. typedef unsigned int UINT;
9. a+b　a-b　a-b
10. iomanip.h　iostream.h

三、编程题

1.

```
#include <iostream>
void main()
{
  int num,var1,var2,var3;
  cout<<"请输入一个三位数："<<endl;
  cin>>num;
  if(num>999||num<100)        //用于检查输入数据的合法性
        cout<<"您的输入有误!"<<endl;
  else
  {
        var1=num/100;
        var2=(num-var1*100)/10;
        var3=num%10;
        cout<<"百位数为："<<var1<<endl
              <<"十位数为："<<var2<<endl
              <<"个位数为："<<var3<<endl;
  }
}
```

2.

```
#include <iostream>
#include <iomanip>
void main()
{
        int array[10];
        enum month{Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec};
        char *p;
        cout<<"The size of char is:"<<sizeof(char)<<endl
              <<"The size of int is:"<<sizeof(int)<<endl
              <<"The size of short int is:"<<sizeof(short int)<<endl
              <<"The size of long int is:"<<sizeof(long int)<<endl
              <<"The size of float is:"<<sizeof(float)<<endl
              <<"The size of double is:"<<sizeof(double)<<endl
              <<"The size of long double is:"<<sizeof(long double)<<endl
              <<"The size of signed int is:"<<sizeof(int)<<endl
              <<"The size of unsigned int is:"<<sizeof(unsigned)<<endl
              <<"The size of array is:"<<sizeof(array)<<endl
              <<"The size of month is:"<<sizeof(month)<<endl
              <<"The size of p is:"<<sizeof(p)<<endl;
  }
```

第 3 章 C++程序的流程控制

【典型例题】

例题 1：下列程序的运行结果为_____。

```
#include <iostream>
void main( )
{
 char c='@' ;
 if (c>='A' && c<='Z')   cout<<"是大写字母 " ;
 else if (c>='a' && c<='z')   cout<<"是小写字母";
    else   cout<<"是其它字符";
 }
```

解答：

本题主要考查 if 语句的嵌套使用方法。首先判断字符变量 c 是否满足 c>='A' && c<='Z'，如果满足则输出"是大写字母 "；否则判断 c 是否满足 c>='a' && c<='z'，如果满足则输出"是小写字母"否则输出"是其它字符"。else 总是与离它最近的前一个 if 配对。

答案为：是其它字符。

例题 2：已定义：char grade; ，若成绩为 A、B、C 时输出合格，成绩为 D 时输出不合格，其他情况提示重新输入。要完成以上功能，则下列 switch 语句正确的是（ ）。

(a)
```
switch(grade){
  case ‘A’:
  case ‘B’:
  case ‘C’: cout<<”合格”;break;
  case ‘D’: cout<<”不合格”;break;
  default: cout<<”请重新输入：”;
}
```

(b)
```
switch(grade){
  case ‘A’:
  case ‘B’:
  case ‘C’: cout<<”合格”;
  case ‘D’: cout<<”不合格”;
 default: cout<<”请重新输入：”;
}
```

(c)
```
switch(grade){
  case‘A’, ‘B’, ‘C’: cout<<”合格”;break;
  case ‘D’: cout<<”不合格”;break;
 default: cout<<”请重新输入：”;
}
```

(d)
```
switch(grade){
  case  A:
  case  B:
  case  C: cout<<”合格”;break;
  case  D: cout<<”不合格”;break;
 default: cout<<”请重新输入：”;
}
```

解答：

本题主要考查 switch 语句的使用。在 switch 语句执行过程中，找到第一个相匹配的表达式后，转去执行该 case 后的语句，直到遇到 break 语句后跳出 switch 语句执行其后的语句。对于选项 b，若 grade 的值为 A 则执行结果为“合格不合格请重新输入”，不满足本题的要求；switch 语句多个 case 分支不能简写为多个表达式之间用逗号隔开的一个 case 分支，选项 c 错误；case 后的表达式只能是整型、字符型或枚举型常量表达式，选项 d 中 case 后的 A、B、C、D 是变量。答案为：a。

例题 3：找出并改正下列程序段中的错误：

```
(1) if(x>0);-------------①
       y=x+1;-------------②
    else;---------------③
       y=x-1;-------------④
(2) While(i) --------------------①
     { cout<<i<<endl; ----------②
        i--;}--------------------③
(3) int i=1,sum; --------①
    while(i<=100)    ------②
    { sum+=i; -----------③
        i++;}-------------④
(4) int i=1,sum=0;-------------①
    do{ ----------------------②
       sum+=i; ----------------③
       i++;}while(i<=100) ------④
```

解答：

本题中包含了初学者在编程中容易犯的一些错误，提醒读者注意。

(1)本题考查对 if 语句的语法结构的掌握。本题的错误在于在 if 和 else 后不应加分号。答案为：将①、③句末的分号去掉。

(2)C++是大小写敏感的语言。答案为：将①中 While 改为 while。

(3)本题目的是完成 1 到 100 求和，结果放在 sum 中，但是 sum 在参与运算前应该首先对其赋值。答案为：将①改为 int i=1,sum=0;

(4)本题主要考查对 do…while 的语法结构的掌握以及与 while 结构进行对比区别两者在作用和语法上的不同。答案为：将第④行 while 后加分号即 while(i<=100);

例题 4：循环语句 for(int i=0;i<=5&&!i;i++) cout<<i<<endl;执行循环次数为（ ）。

(a) 1 次　(b)3 次　(c) 5 次　(d)6 次

解答：

本题考查对 for 循环的理解以及表达式运算。执行 for 循环 i 的初值为 0，第一次循环时表达式 0<=5&&!0 结果为 1 所以执行循环体输出 0；然后 i 自加为 1，计算表达式 1<=5&&!1 结果为 0，所以退出循环。答案为：a。

例题 5：程序段：

```
int i,sum=0;
for(i=1;i<=100;sum+=i,i++);
```

与上边程序段不等价的是（ ）。

```
(a) int sum=0,i=1;
    do{
       sum+=i++;
       }while(i<=100);
(b) int i=1,sum=0;
    while(i<=100)
    {
```

```
        sum+=i++;
    }
(c) int i=1,sum=0;
    while(1)
    {
        if(i>100)
            break;
        sum+=i++;
    }
(d) int i,sum=0;
    for(i=1;i<=100;i++,sum+=i);
```

解答：

本题主要考查对各种循环结构的以及它们之间转换关系的理解。本题中的程序段是求解 1 到 100 的和，循环结束后 i 的值为 101，sum 的值为 5050。选项 a、b 分别用 do…while 和 while 循环完成求解 1 到 100 的和；选项 c 是永真循环，通过 break 语句退出循环，其作用也是求解 1 到 100 的和；选项 d 中 i 先自加然后求和，其作用为求解 2 到 101 的和，循环结束后 i 的值为 101，sum 的值为 5150。答案为：d。

例题 6：运行下列程序的结果为（1）_____。（2）_____。

（1）

```
#include <iostream>
void main()
{
  int x,y,cmultiple;
  cout<<"输入两个整数:";
  cin>>x>>y;
  cmultiple=x;
  while(1)
  {
   if(cmultiple%y==0)break;
   cmultiple+=x;
  }
  cout<<"最小公倍数:"<<cmultiple<<endl;
}
```

输入：24

7

（2）

```
#include <iostream>
void main()
{
  int x,y,var1;
  cout<<"输入两个整数：";
  cin>>x>>y;
  if(x<y)var1=x,x=y,y=var1;
  var1=x%y;
  while(var1)
  {
   x=y;y=var1;var1=x%y;
  }
  cout<<"最大公约数:"<<y<<endl;
}
```

输入：24

7

解答：

本题考查理解程序的能力。(1)中求解两个数 x,y 的最小公倍数思路为若 x 能够被 y 整除则 x 就是这两个数的最小公倍数，否则判断 x 的整数倍是否能被 y 整除，直到 x 的某个倍数可以被 y 整除，则该数即为这两个数的最小公倍数。(2)中求解两个数 x,y 的最大公约数思路为用两个数中较大的数作为被除数，较小的数作为除数，如果能够整除则较小的数就是这两个数的最大公约数，否则，将较小的数作为新的被除数，将两个数的模作为除数，重复以上工作直到模为 0，则这个除数就是 x 和 y 的最大公约数。

答案为：(1) 最小公倍数:168 (2) 最大公约数:1

例题 7：以下程序的功能是输出 1 到 100 之间每位数的乘积大于每位数的和的数，如对数字 12 有 1*2<1+2，所以不输出这个数；对数字 23 有 2*3>2+3 所以输出这个数。请填空。

```
#include <iostream>
void main()
{
        int num,product=1,sum=0,n;
        for(num=1;num<=100;num++)
        {
              product=1;sum=0;
              ________①__n=num__ ______;
              while(_____②__n___)
              {
                      product*=n%10;sum+=n%10;
                      _______③__n/=10____;
              }
              if (product>sum) cout<<num<<endl;
        }
}
```

解答：

本题中变量 num 用于表示 1 到 100 的整数，变量 product 用于存放每位数的乘积，变量 sum 用于存放每位数的和，变量 n 用于存放求解每位数的中间结果。外层 for 循环用于控制判断 1 到 100 的数，内层 while 循环用于计算每位数的积与每位数的和，if 语句用于判断该数的每位数的乘积是否大于每位数的和，若满足此条件，则输出这个数。在①处要对 n 赋值，最初它是 num 的副本而后用于存放计算每位数的中间结果；②处为退出 while 循环的条件，当当前这个数的每位数都被取出，则内层循环结束；③改变循环变量的语句，使得循环在某种情况下可以结束，这里取 n 除 10 的商，直到商为 0 时结束内层循环。

答案为：① ② ③

例题 8：以下程序的功能是判断一个数是否为素数。请填空。

```
#include <iostream>
void main()
{
        int num;
        cout<<"输入一个正整数：";
        _________①________;
        int isprime=1;
        for(int i=2;i<=num-1;i++)
              if(________②________)
              {
                      isprime=0;
```

```
                    _____③_____;
                }
        if(isprime)
                cout<<num<<" 是一个素数。"<<endl;
        else
                cout<<num<<" 不是一个素数。"<<endl;
}
```

解答：

本题中变量 num 存放要判断的数，变量 isprime 用于记录该数是否为素数，当 isprime 为 1 时即该数为素数，否则为合数。判断思路为如果 num 能被 2 到 num-1 的任意一个数整除则该数不是素数。①处需要输入待判断的数，②处为判断条件，当检测到 2 到 num-1 中第一个能整除 num 的数时则可判断出该数不是素数，此时退出循环，故③为退出语句。

答案为： ①cin>>num　　②num%i==0　　③break

例题 9：编写一个程序，输入一个正整数，判断它是否能被 3，5，7 同时整除。

解答：

参考程序如下

```
#include <iostream>
void main()
{
        int num;
        cout<<"请输入一个正整数：";
        cin>>num;
        if(num<0)
                cout<<"输入有误!";
        else
                if(num%3==0&&num%5==0&&num%7==0)
                        cout<<num<<" 能被 3、5、7 同时整除。"<<endl;
                else
                        cout<<num<<" 不能被 3、5、7 同时整除。"<<endl;

}
```

例题 10：编写一个程序，让用户输入年和月，然后判断该月有多少天。

解答：

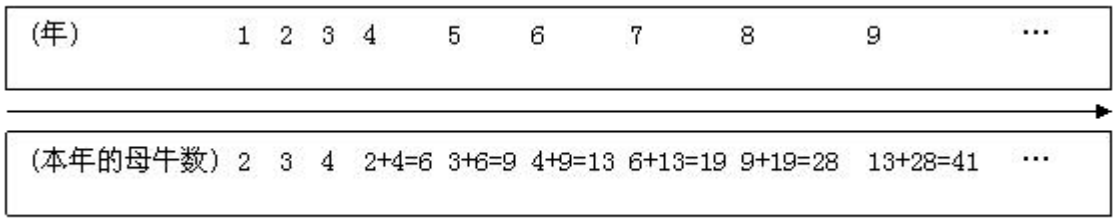
算法思想：判断某年某月有多少天，每个月的天数有四种可能：1、3、5、7、8、10、12 月为 31 天，4、6、9、11 月为 30 天，闰年的 2 月有 29 天，不是闰年则 2 月为 28 天。因为每月的天数有多种可能，我们选择用 switch 语句解决本题。程序流程是首先输入要判断的年月，然后判断该年月应有多少天，最后输出结果。

参考程序如下：

```
#include <iostream>
void main()
{
  int year,month,days,leap;
  cout<<"请输入年月:";
  cin>>year>>month;
  switch(month)
  {
   case 1:
   case 3:
   case 5:
```

```
  case 7:
  case 8:
  case 10:
  case 12:days=31;break;
  case 4:
  case 6:
  case 9:
  case 11:days=30;break;
  case 2:if(year%400==0)
         leap=1;
       else if(year%4==0&&year%100!=0)
              leap=1;
            else leap=0;
       if(leap)   days=29;
       else     days=28;
  }
   cout<<year<<"年"<<month<<"月的天数为:"<<days<<endl;
}
```

例题 11：编写一个程序。计算若一头母牛，它每年年初生一头小母牛，每头小母牛从出生起第四个年头开始每年也生一头小母牛，按此规律，第 10 年时有多少头母牛？
解答：
算法思想：



假设 f(n)表示第 n 年的母牛数，已知 f(0)=0,f(1)=2,f(2)=3,f(3)=4,推得在第 n 年时应有 f(n-3)头母牛生育出 f(n-3)头母牛，所以第 n 年共有 f(n-1)+f(n-3)头母牛。据此得出数学表达式：f(n)=f(n-1)+f(n-3)。在下边的参考程序中分别用变量 sum、sum1、sum2、sum3 表示 f(n),f(n-1),f(n-2),f(n-3)其中 n 从 4 变化到 10。

参考程序如下：

```
#include <iostream>
void main()
{
        int sum1=2,sum2=3,sum3=4,sum=0,n=10;
        for(int i=4;i<=n;i++)
        {
                sum=sum1+sum3;
                sum1=sum2;
                sum2=sum3;
                sum3=sum;
        }
          cout<<"第十年有"<<sum<<"头牛!"<<endl;
}
```

例题 12：计算 $e=1+\frac{1}{1!}+\frac{1}{2!}+\ldots+\frac{1}{n!}+\ldots$，当通项 $\frac{1}{n!}<10^{-7}$ 时停止计算。

解答：

算法思想：

本题主要考查学生对循环结构的运用以及对数学问题编程技巧的掌握。本题为有规律的若干项相加，所以采用循环结构处理。循环的条件是当 $\frac{1}{n!}\geq 10^{-7}$ 时继续运算直到这个条件不满足时就达到了题目要求的精度，则停止运算。循环体中完成三件事，首先是求出本次（第 i 次）循环中的 $\frac{1}{i!}$ 的值，然后计算当前的 e，最后完成循环变量增 1 的操作。在求解本题时主要注意计算中所采用的变量的数据类型以及如何完成运算精度的控制。

参考程序如下：

```
#include  <iostream>
int main()
{
            double e=1.0;
            double x=1.0;
            int i=1;
            while(x>=1.0e-7)
            {
              x=x/i;
              e=e+x;
              i=i+1;
            }
            cout << "e = " << e<<endl;
            return 0;
}
```

【习题】

一、 选择题

1. 在循环语句中使用 break 语句的作用是（ ）。
   (a)结束本次循环　　(b)结束该层循环　　(c)结束所有循环　　(d)结束程序执行
2. 对 if 后的括号中的表达式，要求 i 不为 0 的时候表达式为真，该表达式表示正确的为（ ）。
   (a)i　　(b)!i　　(c)i<>0　　(d)i=0
3. 下列循环语句的执行次数是（ ）。
   while(!1) cout<<”ok!”;
   (a)0 次　　(b)1 次　　(c)2 次　　(d) 无数次
4. 运行下列程序结果为（ ）。

```
#include <iostream>
void main()
{
       int i;
       for (i=0;i<=10;i++){
              if (i%2) cout<<i;
              else continue;
```

```
            }
    }
```

(a)246810　　(b)12345　　(c)678910　　(d)13579

二、　填空题

1. 结构化程序设计的三种基本结构是_____、_____、_____。

2. continue 语句实现的作用是_____。

3. 若输入”china 2008!”,运行下列程序的输出结果为_____。

```
#include  <iostream>
#include <stdio.h>
void main( )
{
       char c;
       int letters=0,digits=0,others=0;
       cout<<"Please input a line charaters"<<endl;
       while ((c=getchar( ))!='\n')
       {
              if (c>='a' && c<='z' || c>='A' && c<='Z' )
                     letters++;
              else
                     if (c>='0' && c<='9')
                            digits++;
                     else
                            others++;
       }
       cout<<"letters:"<<letters<<endl
              <<"digits"<<digits<<endl
              <<"others"<<others<<endl;
}
```

4. 本程序完成 1!+2! +……+18!的计算，请将程序补充完整。

```
#include <iostream>
#include <iomanip>
void main()
{
       double sum=0,fac=1;
       for(int i=1;i<=18;i++)
       {
              _____①_____

              _____②_____
       }
       cout<<"1!+2!+……+18!="
              <<setiosflags(ios::fixed)<<setprecision(0)
              <<sum<<setprecision(6)<<endl;
}
```

三、　编程题

1. 输入某学生成绩，若成绩在 90-100 输出＂优秀＂，若成绩在 80-89 输出＂良好＂，若成绩在 70-79 输出＂中＂，若成绩在 60-69 输出”及格”，若成绩在 0-59 输出”不及格”。

2. 输入三人数，按从小到大的大顺序输出。

3. 在 100～200 中找出同时满足用 3 除余 2，用 5 除余 3 和用 7 除余 2 的所有整数。
4. 求 100～999 中的。所谓水仙花数是指一个三位数，它的每位数字的立方之和等于该数。例如，因为 153＝1^3＋5^3＋3^3，所以 153 为水仙花数。
5. 求 1000 之内的所有完数。所谓完数是指一个数恰好等于它的所有因子之和。例如，6＝1＋2＋3，所以 6 为完数。
6. 编一程序显示如下图案：

```
    *
   * * *
  * * * * *
 * * * * * * *
* * * * * * * * *
 * * * * * * *
  * * * * *
   * * *
    *
```

7. 编一程序显示如下图案：

```
    A
   A B C
  A B C D E
 A B C D E F G
```

8. 猴子吃桃问题。猴子第一天摘下若干个桃子，当即吃了一半，还不过瘾，又多吃了一个。第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第 10 天早上想再吃时，发现只剩一个桃子了，求猴子第一天究竟摘了多少个桃子？
9. 编程序模拟剪刀，石头和纸游戏。游戏规则为：剪刀剪纸，石头砸剪刀，纸包石头．玩游戏者从键盘上输入Ｓ（表示剪刀）或Ｒ（表示石头）或Ｐ（表示纸），要求两个游戏者交替输入，计算机给出输赢的信息。
10. 编写程序输出菲波那切数列的前 20 项。即前两项为 1，以后每一项为前两项之和。
11. 打印九九乘法表。

【参考答案】

一、选择题

1-4. baad

二、填空题

1．顺序结构、选择结构（分支结构）、循环结构

2．跳出本次循环。

3．

```
please input a line charaters
letters:5
digits:4
others:2
```

4．①fac*=i;　②sum+=fac;

三、编程题

1.

```
#include <iostream>
void main()
{
	double grade;
	char* degree;
	cout<<"请输入学生成绩：";
	cin>>grade;
	if(grade>100||grade<0)
	{
		cout<<"您的输入有误!"<<endl;
		return;
	}
	else
		if(grade>=70)
			if(grade<80)
				degree="中";
			else if(grade<90)
					degree="良好";
				 else
					degree="优秀";
		else if(grade>=60)
				degree="及格";
			 else
				degree="不及格";

	cout<<"分数:"<<grade<<endl
		<<degree<<endl;
}
```

2.

```
#include <iostream>
void main()
{
	int num1,num2,num3,num;
	cout<<"请输入三个整数：";
	cin>>num1>>num2>>num3;
	if(num1>num2)
	{
		num=num1;
		num1=num2;
		num2=num;
	}
	if(num1>num3)
	{
		num=num1;
		num1=num3;
		num3=num;
	}
	if(num2>num3)
	{
		num=num2;
		num2=num3;
		num3=num;
	}
```

```
    cout<<"三个数按从小到大输出为："<<endl
        <<num1<<endl
        <<num2<<endl
        <<num3<<endl;
}
```

3.

```
#include <iostream>
void main()
{
    cout<<"在 100～200 中同时满足用 3 除余 2，用 5 除余 3 和用 7 除余 2 的整数为："<<endl;
    for(int i=100;i<=200;i++)
    {
        if(i%3==2&&i%5==3&&i%7==2)
            cout<<i<<endl;
    }
}
```

4.

```
#include <iostream>
#include<math.h>
void main()
{
    int x,y,z,sum;
    cout<<"100～999 中的水仙花数为："<<endl;
    for(int i=100;i<=999;i++)
    {
        x=i/100;
        y=i%100/10;
        z=i%10;
        sum=pow(x,3)+pow(y,3)+pow(z,3);
        if(i==sum)
            cout<<i<<endl;
    }
}
```

5.

```
#include <iostream>
void main()
{
    int sum;
    cout<<"1000 之内的所有完数为："<<endl;
    for(int i=1;i<=1000;i++)
    {
        sum=0;
        for(int j=1;j<=i/2;j++)
            if(i%j==0)
                sum+=j;
        if(i==sum)
            cout<<i<<endl;
    }
}
```

6.

```
#include <iostream>
void main()
```

```
{
    int i,j,n;
    cout<<"请输入上三角行数：";
    cin>>n;
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=(n-i)*2;j++)
            cout<<' ';
        for(j=1;j<=2*i-1;j++)
            cout<<"* ";
        cout<<endl;
    }
    for(i=n-1;i>=1;i--)
    {
        for(j=1;j<=(n-i)*2;j++)
            cout<<' ';
        for(j=1;j<=2*i-1;j++)
            cout<<"* ";
        cout<<endl;
    }
}
```

7.

```
#include <iostream>
#include <iomanip>
void main()
{
    int i,j;
    char x;
    for(i=1;i<=4;i++)
    {
        x='A';
        for(j=1;j<=2*(4-i);j++)
            cout<<' ';
        for(j=1;j<=2*i-1;j++)
        {
            cout<<setw(2)<<x;
            x+=1;
        }
        cout<<endl;
    }
}
```

8.

```
#include <iostream>
void main()
{
    int peach=1;
    for(int day=1;day<10;day++) //day 表示 n 天前
        peach=2*(peach+1);
    cout<<"猴子第一天摘了"<<peach<<"个桃子！";
}
```

9.

```
#include <iostream>
void main()
{
```

```
    char play1,play2;
    char* result;
    cout<<"请两位玩家顺序输入Ｓ（表示剪刀）或Ｒ（表示石头）或Ｐ（表示纸）：";
    cin>>play1>>play2;
    if((play1=='R'||play1=='S'||play1=='P')&&(play2=='R'||play2=='S'||play2=='P'))
    {
        switch(play1)
        {
        case 'S':
            if(play2=='R') result="玩家 2 赢！";
            else if(play2=='P') result="玩家 1 赢！";
                else result="平局！";
            break;
        case 'R':
            if(play2=='P') result="玩家 2 赢！";
            else if(play2=='S') result="玩家 1 赢！";
                else result="平局！";
            break;
        case 'P':
            if(play2=='S') result="玩家 2 赢！";
            else if(play2=='R') result="玩家 1 赢！";
                else result="平局！";
            break;
        }
        cout<<result<<endl;
    }
    else
        cout<<"输入有误!"<<endl;
}
10．
#include <iostream>
void main()
{
    int item,item1=1,item2=1;
    cout<<"菲波那切数列的前 20 项为："<<endl;
    cout<<item1<<endl
        <<item2<<endl;;
    for(int i=3;i<=20;i++)
    {
        item=item1+item2;
        cout<<item<<endl;
        item1=item2;
        item2=item;
    }
}
11．
#include <iostream>
#include <iomanip>
void main()
{
    cout<<setw(3)<<'*';
    for(int title=1;title<=9;title++)
        cout<<setw(3)<<title;
```

```
    cout<<endl;
    for(int row=1;row<=9;row++)
    {
        cout<<setw(3)<<row;
        for(int line=1;line<=row;line++)
            cout<<setw(3)<<row*line;
        cout<<endl;
    }
}
```

第 4 章 数组

【典型例题】

例题 1：下面定义数组的语句正确的是（ ）。
(a) int i=6;char a[i]= “hello”;
(b) const int i=5;char a[i]= “hello”;
(c) char a[6]=”hello”;
(d) char a5=”hello”;
解答：
本题主要考查字符数组在定义及初始化时需要注意的问题。数组定义中数组长度不能指定为除 const 变量以外的变量，选项 a 中 i 为变量；选项 b 中字符串“hello”含有 6 个字符，而数组 a 的长度是 5 所以编译器报错；选项 d 中 a5 是一个字符变量而不是字符数组，而“hello”含有 6 个字符，无法放在一个字符变量中。所以答案为：c

例题 2：已知数组 a 定义为：int a[][3]={{1,2,3},{4}};， 则 a[1][2]的值为（ ）。
(a)2 (b)3 (c)4 (d)0
解答：
在对数组进行初始化时只给出部分元素的初始值，则剩余元素自动初始化为 0。答案为：d。

例题 3：已知数组定义为 int a[2][4];，下列对数组元素引用正确的为（ ）。
(a)a[1,2] (b)a[1][2] (c)a(1,2) (d)a[1][4]
解答：
对二维数组的引用格式为：数组名[下标 1][下标 2]；n 行 m 列的二维数组其第一维的下标从 0 到 n-1，第二维的下标从 0 到 m-1。选项 a 和 c 引用格式错误，d 中下标越界。
答案为：b。

例题 4：下列程序段错误的是（ ）。
(a)
```
char str1[8];
cin>>str1;
```
(b)
```
char str1[8];
strcpy(str1,”first”);
```
(c)
```
char str1[8];
for(int i=0;i<7;i++)
    cin>>str1[i];
str1[7]=’\0’;
```

(d) char str1[8],str2[8]=”first”;
str1=str2;

解答：

本题主要考查如何为字符数组赋值。为字符数组赋值可以直接从键盘输入一个字符串、用strcpy函数将一个字符串复制到该字符数组或者用循环语句逐个为字符数组元素赋值。不能直接将一个数组赋值给另一个数组。答案为：d。

例题5：下列说法正确的是（ ）。

(a)数组可以存放不同类型的元素。 (b)定义int a[2];,则数组a有两个数组元素。

(c)定义int a[3];,则该数组中元素分别为：a[0],a[1],a[2],a[3]。

(d)在编译时，不必确定数组的大小。

解答：

数组中所有的元素具有相同类型，选项a错误；定义int a[3];则数组a中含有三个元素分别为a[0],a[1],a[2]，选项c错误；编译时必须确定数组的大小，即在定义数组时必须给出数组长度。

答案为：b。

例题6：运行下列程序结果为_____。

```
#include <iostream>
#include <iomanip>
void main()
{
    int array1[3][3]={{1,2,13},{4,5,16},{7,8,9}},i=0,j=2,sum1=0,sum2=0;
    for(;i<3;i++)
    {
        sum1+=array1[i][i];
        sum2+=array1[i][j--];
    }
    cout<<sum1<<setw(5)<<sum2<<endl;
}
```

解答：

本题中程序的作用是对一个给定的3×3矩阵求出其主对角线元素值之和与逆对角线元素值之和然后输出计算结果。其中主对角线上的元素即为行和列下标相同的元素，逆对角线元素下标满足行下标从小到大依次递增同时列下标从大到小依次递减的元素。答案为：15 25。

例题7：下列程序的作用是检查字符串s中是否包含字符串t，若包含，则返回并输出t在s中的开始位置（下标值），否则返回-1。请将程序补充完整。

```
#include <iostream>
int main()
{
    int i,j,k;
    char s[20]="Today is sunday!",t[10]="sun";
    for(i=0;s[i]!='\0';i++)
    {
        for(j=i,k=0;_____①_____ &&s[j] == t[k];j++,k++);
        if(_____②_____)
        {
            cout<<"t在s中的开始位置下标为:"<<i<<endl;
            return i;
        }
    }
```

```
    return -1;
}
```
解答：
本程序用于解决检查一个字符串是否包含于另一个字符串。程序的思想是从 s 串的第 0 个字符起与 t 串进行比较，若到 t 串结束时所比字符均相等则说明 t 在 s 中，返回 t 在 s 中的开始位置，程序结束；否则从 s 的第 1 个字符起重复以上操作，依此类推直到匹配或 s 串结束为止。程序中变量 i 用于记录本轮中正在与 t 串相比较的 s 串的子串的首字符位置（下标），k 用于记录 t 串中当前比较字符的位置(下标),j 用于记录 s 串中当前比较的字符的位置(下标)。因此，答案为：①t[k]!='\0' ②t[k]=='\0'

例题 8：编程实现任意输入 10 个数，然后按从小到大的顺序输出这 10 个数。
解答：
算法思想：关于排序问题的实现算法非常多。有插入排序、希尔排序、快速排序、选择排序、归并排序、基数排序等。在时间空间复杂度方面这些算法各有优缺点。有关这方面的内容可以参考数据结构方面的介绍。在这里给出了一个采用简单选择排序的参考程序。其思路为循环 n-1 次（n 为元素个数），每次循环都从剩下的数中选择出最小的一个。如下给出排序过程：

下标 i	0	1	2	3
list[i]	100	34	20	200
第 1 轮	***34***	***100***	20	200
	20	100	***34***	200
第 2 轮	20	***34***	***100***	200
第 3 轮	20	34	100	200

参考程序：
```
#include  <iostream>
#include <iomanip>
int main()
{
    const int COUNT=10;
    int list[COUNT];
    int i,j,tmp;
    cout<<"请输入 10 个数："<<endl;

    for(i=0;i<COUNT;i++)   //输入数组元素
        cin>>list[i];

    for(i=0;i<COUNT-1;i++)        //排序
    {
        for(j=i+1;j<COUNT;j++)
            if(list[j]<list[i])
            {
                tmp=list[i];
                list[i]=list[j];
                list[j]=tmp;
            }
    }

    for(i=0;i<COUNT;i++)         //输出排序后的结果
        cout<<setw(5)<<list[i];
    cout<<endl;
```

```
    return 0;
}
```

【习题】

一、选择题

1.在C++中对数组下标说法正确的是（　）。

(a)初始化数组的值的个数可以多于定义的数组元素的个数，多出部分将被忽略。

(b)初始化数组的值的个数可以少于定义的数组元素的个数。

(c)初始化数组的值的个数必须等于定义的数组元素的个数。

(d)初始化数组的值可以通过跳过逗号的方式来省略。如 int a[3]={1,,2};

2.数组定义为:int a[2][2]={1,2,3,4};则 a[1][0]%3 为（　）。

(a)0　　(b)1　　(c)2　　(d)4

3.数组定义为：int a[][2]={5,6,1,2,3,8};则能用于计算数组下标的是（　）。

(a)sizeof(a)/sizeof(int)　　(b)sizeof(a[])/sizeof(3)

(c)sizeof(a[][2])/sizeof(int)　　(d)sizeof(a)/sizeof(a[2][1])

4.运行下列程序结果为（　）。

```
#include <iostream>
void main()
{
int a[4]={1,2,3,4};
for(int i=3;i>=0;i--)
    cout<<a[i];
}
```

(a)1234　　(b)1324　　(c)4231　　(d)4321

5.运行下列程序结果为（　）。

```
#include <iostream>
void main()
{
    int i,j,t,a[2][2]={8,7,6,5};
    for(i=0;i<1;i++)
        for(j=i+1;j<2;j++)
        {
            t=a[i][j];
            a[i][j]=a[j][i];
            a[j][i]=t;
        }
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
            cout<<a[i][j];
        cout<<endl;
    }
}
```

(a)87　　(b)78　　(c)86　　(d)68

65　　65　　75　　57

二、填空题

1. $m \times n$ 数组包含_____行、_____列和_____个元素。

2. 定义数组 int a[10];，若要给该数组的第三个元素赋值 100，其语句为_____。

3. 已知数组 a 中的元素个数为 4，下列语句的作用是将下标为 i 的元素移动到下标为 i－1 的单元，其中 1≤i＜4。a 中原有数据为 1，2，3，4，移动后 a 中元素结果为 2,3,4,4。请将下列程序补充完整。

```
#include <iostream>
void main()
{
 int a[4]={1,2,3,4};
 for(int i=0;_____①_____;i++)
     a[i]=_____②_____;
}
```

4. 程序填空

运行下列程序后当 str 是对称的时，输出“是回文”，否则输出“不是回文”，请将程序补充完整。

```
#include <iostream>
void main()
{
   char str[20];
   cin.get(str,20);//输入字符串
   int i=0, j=0;
   while (str[j])_____①_____;
   for(j--; i<j && str[i]==str[j]; i++,j--);
   if(_____②_____)cout<<"是回文";
   else
      cout<<"不是回文";
}
```

5. 运行下列程序的结果为_____。

```
#include <iostream>
#include <iomanip>
void main()
{
 int array1[3][3]={1,2,3,4,5,6,7,8,9},array2[3][3],i,j;
 for(i=0;i<3;i++)
     for(j=0;j<3;j++)
         array2[j][i]=array1[i][j];
 for(i=0;i<3;i++)
 {
     for(j=0;j<3;j++)
         cout<<setw(3)<<array2[i][j];
     cout<<endl;
 }
}
```

6. 运行下列程序的结果为_____。

```
#include <iostream>
void main()
{
 int num[6],i,j;
 cout<<"请输入 1 到 50 的六个正整数：";
   for(i=0;i<6;i++)
      cin>>num[i];
 for(i=0;i<6;i++)
```

```
    {
        for(j=1;j<=num[i];j++)
            cout<<"*";
        cout<<endl;
    }
    }
```

输入为：2 4 1 6 3 1

三、编程题

1. 编写一个程序实现矩阵的乘法运算。
2. 输入一个 4×4 矩阵各元素的值，求解该矩阵中的马鞍点（即该点的值在它所在的行中最大，在它所在的列中最小）。
3. 写统计输入的正文中有多少单词的程序，这里的单词指的是用空白符分隔开的字符串。
4. 编写程序实现一个简单的加密器，实现英文字符串的加密。加密规则如下：将字符替换成它后面的第三个字符。如”abc”换成”def”。
5. 编写一程序，将字符数组 s2 中的全部字符拷贝到字符数组 s1 中。不用 strcpy 函数。拷贝时，'\0'也要拷贝过去。'\0'后面的字符不拷贝。
6. 有 17 个人围成一个圈(编号 0-16)，从第 0 号的人开始从 1 报数，凡报到 3 的倍数的人离开圈子，然后再继续数下去。直到最后只剩下一个人为止。问此人原来的位置是多少号？
7. 给定一个升序数组，该数组的元素值为 1，3，5，7，9，11，13，任意输入一个数判断该数在数组中是否存在。若存在，给出它在数组中的位置，否则显示该数不存在。
8. 打印输出杨辉三角形（共输出 10 行）。

【参考答案】

一、选择题

1-5. baadc

二、填空题

1．m　　n　　m×n

2. a[2]=100;

3．①i<3　　②a[i+1]

4．①j++ ②i==j

5．

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

6．

```
 **
****
*
******
***
*
```

三、编程题

1．

```
#include <iostream>
#include <iomanip>
void main()
{
```

```
    const int A1=2,A2=3,A3=2;
    int i,j,k;
    double array1[A1][A2]={{1.0,2.0,3.0},{4.0,5.0,6.0}},
        array2[A2][A3]={{1.0,1.0},{1.0,1.0},{1.0,1.0}},
        array3[A1][A3]={0,0,0,0};
    for(i=0;i<A1;i++)
        for(j=0;j<A3;j++)
            for(k=0;k<A2;k++)
                array3[i][j]+=array1[i][k]*array2[k][j];
    for(i=0;i<A1;i++)
    {
        for(j=0;j<A3;j++)
            cout<<setw(4)<<array3[i][j];
        cout<<endl;
    }
}
```

2.

```
#include <iostream>
void main()
{
    int array[4][4],i,j,k,max,col,flag=0;
    cout<<"请输入 4 行 4 列矩阵：";
    for(i=0;i<4;i++)
        for(j=0;j<4;j++)
            cin>>array[i][j];
    for(i=0;i<4;i++)
    {
        max=array[i][0];col=0;
        for(j=1;j<4;j++)
        {
            if(array[i][j]>max)
            {
                max=array[i][j];
                col=j;
            }
        }
        for(k=0;k<4;k++)
        {
            if(array[k][col]<=max&&k!=i)
                break;
        }
        if(k==4)
        {
            cout<<"马鞍点在"<<i<<"行"<<col<<"列:"<<array[i][col];
            flag=1;
        }
    }
    if(!flag)
        cout<<"这个矩阵中没有马鞍点！";
}
```

3.

```
#include <iostream>
void main()
{
```

```
	const int MAX=100;
	char str[MAX];
	int i,num=0;
	cin.get(str,100);//输入字符串，以回车结束
	for(i=0;i<MAX&&str[i]!='\0';i++)
	{
		if(str[i]==' ')
			num++;
	}
	if(num!=0)num++;
	cout<<"单词数为："<<num<<endl;
}
```

4.

```
#include	<iostream>
void main()
{
	const LENGTH = 100;
	char str[LENGTH];
	cout<<"请输入字符串:";
	cin>>str;
	for(int i=0; i<LENGTH;i++)
	{
		if(str[i] == '\0') break;
		str[i] = str[i] + 3;
	}
	cout<<"加密后的字符串为："<<str<<endl;
}
```

5.

```
#include <iostream>
void main()
{
	int i,j,len1=0,len2=0;
	char s1[20]="hello ",s2[10]="world";
	for(i=0;i<20;i++)
		if(s1[i]!='\0')
			len1++;
		else
			break;
	for(i=0;i<10;i++)
		if(s2[i]!='\0')
			len2++;
		else
			break;
	if((len1+len2)>19)
	{
		cout<<"存储空间不足!";
		return;
	}
	i=len1;
	j=0;
	while(i<20&&s2[j]!='\0')
		s1[i++]=s2[j++];
	s1[i]='\0';
	cout<<s1<<endl;
```

```
}
```
6.
```
#include <iostream>
void main()
{
	int array[17],n=17,i,j=1;
	for(i=0;i<17;i++)
		array[i]=1;
	i=1;
	while(n!=1)
	{
		if(array[(j-1)%17]==0)
		{
			j++;continue;
		}

		if(i%3==0)
		{
			array[(j-1)%17]=0;
			n--;
		}
		j++;i++;
	}
	for(i=0;i<17;i++)
	{
		if(array[i]==1) cout<<"最后一个人原来的位置（下标）是:"<<i<<endl;
	}
}
```
7.
```
#include <iostream>
void main()
{
	int array[7]={1,3,5,7,9,11,13},x,low=0,high=6,mid,flag=0;
	cout<<"请输入待查找的数："<<endl;
	cin>>x;
	while(low<=high)
	{
	mid=(high+low)/2;
	if(x==array[mid])
	{
		cout<<"找到了，该数的下标为："<<mid<<endl;
		flag=1;
		break;
	}
	else
		if(x<array[mid])
			high=mid-1;
		else
			low=mid+1;
	}
	if(!flag)
		cout<<"该数不存在！"<<endl;
}
```
8.

```
#include <iostream>
#include <iomanip>
void main()
{
	const int ROW=10;
	const int COL=10;
	int yh[ROW][COL],row,col;
	for(row=0;row<ROW;row++)
	{
		yh[row][0]=1;yh[row][row]=1;
	}
	for(row=2;row<ROW;row++)
	{
		for(col=1;col<row;col++)
		{
			yh[row][col]=yh[row-1][col-1]+yh[row-1][col];
		}
	}
	for(row=0;row<ROW;row++)
	{
		for(col=0;col<=row;col++)
			cout<<setw(5)<<yh[row][col];
		cout<<endl;
	}
}
```

第 5 章 C++函数与程序结构

【典型例题】

例题 1：下列函数定义语句正确的是（ ）。

(a)

```
void fun1(int i,int j);
{
    cout<<i+j;
}
```

(b)

```
void fun1(int i,int j)
{
    void fun2()
    {
        cout<<”This is fun2.”<<endl;
    }
    cout<<i+j;
}
```

(c)

```
void fun1(int i,int j)
{
    cout<<i+j;
    return 1;
}
```

(d)

```
void fun1(int i,int j)
{
    cout<<i+j;
}
```

解答：

本题主要考查对函数定义方法的掌握。函数定义的形式为

```
    函数类型  函数名（形式参数列表）
            {
                函数体
            }
```

选项 a 中多了一个分号；函数可以嵌套调用但是不能嵌套定义，选项 b 在函数内部定义函数是错误的；若果函数类型定义为 void 型，则该函数没有返回值，选项 c 定义了 void 型函数却有返回值，这是错误的。答案为：d

例题 2：下列关于 C++函数的叙述中，正确的是（ ）。

(a)每个函数至少要具有一个参数

(b)每个函数都必须返回一个值

(c)函数在被调用之前必须先声明或定义

(d)函数不能自己调用自己

解答：
本题主要考查对函数的要素及其调用方法的理解。函数可以有参数，也可以没有参数（无参函数），选项 a 错误；函数可以有返回值，也可以没有返回值，当没有返回值时将这个函数定义为 void 型，选项 b 错误；如果一个函数自己调用自己则称为递归调用，这是允许的。选项 d 错误。函数在调用之前必须已经声明或定义过。答案为：c。

例题 3：下面的函数声明语句正确的是（　）。
(a) int fun(int var1=1,char* var2=”Beijing”,double var3);
(b) int fun(int,char* =”Beijing”,double =3.14159);
(c) int fun(int var1=1,char* var2=”Beijing”,double var3=3.14159);
　　int fun(int,char*,double var3= 12.34);
(d) int fun(int var1=1,char*,double var3=3.14159);

解答：
本题主要考查带默认参数的函数原型声明方法。函数调用时实参与形参按照从左到右顺序匹配，在对默认值进行定义时应该从右向左定义。选项 a 和 d 都没有遵从对默认值定义时应该从右向左定义的原则，即对于第一个有默认值的参数而言，它后面还有参数没有定义默认值这是错误的。选项 c 对函数中第 3 个参数定义了两次，错误。答案为：b
本题考查默认参数问题。

例题 4：运行下列程序结果为_____。
```
#include  <iostream>
int f(int[ ],int);
void main()
{
    int a[]={-1,3,5,-7,9,-11};
    cout<<f(a,6)<<endl;
}
int f(int a[],int size)
{
    int i,t=1;
    for(i=0;i<size;i++)
          if(a[i]>0) t*=a[i];
    return t;
}
```
解答：
本题主要考查对数组参数的理解与应用。本程序的作用是计算数组中元素值为正数的元素的乘积。函数 f 含有两个形式参数，第一个参数存放一个整型数组的首地址，第二个参数存放该数组的长度，在该函数中仍然通过数组名[下标]的方式对数组元素进行引用。在主函数中定义了一个数组 a，然后调用函数 f，传递实参时第一个实参 a（数组名）为数组 a 的首地址，第二个参数 6 为数组 a 的长度。答案为：135

例题 5：运行下列程序结果为（　）。
```
#include<string.h>
#include <iostream>
void main( )
{
    char str[][10]={"vb","pascal","c++"},s[10];
    strcpy(s,(strcmp(str[0],str[1])<0?str[0]:str[1]));
    if (strcmp(str[2],s)<0) strcpy(s,str[2]);
    cout<<s<<endl;
```

```
}
```
解答：
本程序的作用是将"vb","pascal","c++"三个字符串中最小的一个放入数组 s，并输出 s。采用二维数组 str 存放三个字符串，则 str[0]、str[1]、str[2]中分别存放三个字符串的首地址，strcmp()函数用于比较两个字符串的大小，strcpy()函数用于将第二个参数所指字符串复制到第一个参数所指数组。C++中有一些操作字符串的标准的库函数，头文件 string.h 包含所有字符串处理函数的说明。常用的一些函数如下：

```
strcpy(char destination[], const char source[]);
//把第二个字符串拷贝到第一个字符串
strncpy(char destination[], const char source[], int numchars);
strcat(char target[], const char source[]);
strncat(char target[], const char source[], int numchars);
int strcmp(const char firststring[], const char secondstring);
//比较两个字符串是否相等
strlen( const char string[] );//返回字符串长度
```

本题答案为：c++

例题 6：运行下列程序的结果为_____。

```
#include <iostream>
void fun1(const double& i)
{
    cout<<i<<endl;
}
void fun2(double &j)
{
    cout<<j<<endl;
}
void fun3(double k)
{
    cout<<k<<endl;
}
void main()
{
    fun1('a');
    fun2('a');
    fun3('a');
}
```

(a)97　　(b)97　　(c)a　　(d)程序出错，无法运行。
　97　　　a　　　a
　97　　　97　　　a

解答：
文字量、常量和需要类型转换的参数都可以传递给 const&参数，但不能传递给非 const 的引用参数。也就是说对非 const 引用参数不允许做类型转换。本题中对于函数 fun1 以及 fun3 在调用时参数都发生了隐形数据类型转换，而对于函数 fun2 它的参数为非 const 的引用参数，不允许作类型转换，所以对 fun2 的调用出错。因此程序出错无法运行。答案为：d。

例题 7：运行下列程序的结果为_____。

```
#include <iostream>
#include <iomanip>
int findmax(int Iarg[]);
float findmax(float Farg[]);
```

```
double findmax(double Darg[]);

main()
{
  int Iarg[6]={15,88,34,12,31,10};
  float Farg[6]={145.5,32.3,363.2,19.3,70.1,35.4};
  double Darg[6]={15.54323,2.47763,63.29876,19.67863,78.34541,35.44009};
  cout<<"largest value in the Iarg is "<<(findmax(Iarg))<<"\n";
  cout<<"largest value in the Farg is "<<(findmax(Farg))<<"\n";
  cout<<"largest value in the Darg is "<<(findmax(Darg))<<"\n";
  return 0;
}
int findmax(int Iarg[])
{
   int max=0;
   for(int i=0;i<6;i++)
   {
     if(Iarg[i]>max)
     {
       max=Iarg[i];
     }
    }
return max;
}
float findmax(float Farg[])
{
   float max=0;
   for(int i=0;i<6;i++)
   {
     if(Farg[i]>max)
     {
       max=Farg[i];
     }
    }
return max;
}
double findmax(double Darg[])
{
   double max=0;
   for(int i=0;i<6;i++)
   {
     if(Darg[i]>max)
     {
       max=Darg[i];
     }
    }
return max;
}
```

解答：

本题主要考查函数重载。本程序用于求解并输出一个整型数组的最大元素、一个浮点型数组的最大元素和一个双精度型数组的最大元素。在主函数中调用三次 findmax()函数，根据传

递实参数组类型不同，自动调用不同的函数完成求解操作，这里 findmax()函数实现了重载。
答案为：

```
largest value in the Iarg is 88
largest value in the Farg is 363.2
largest value in the Darg is 78.3454
```

例题 8：运行下列程序的结果为_____。

```
#include <iostream>
int fun(int,int);
void main()
{
  cout<<"n="<<fun(0,0)<<endl;
}
int fun(int n,int s)
{
   int s1,n1;
   s1=s+n*n;
   if(s1<100)
   {
     n1=n+1;
     fun(n1,s1);
   }
   else
    return n-1;
}
```

解答：
本题主要考查对递归调用程序的理解。

f	s1	n1
f(7,91)		
f(6,55)	91	7
f(5,30)	55	6
f(4,14)	30	5
f(3,5)	14	4
f(2,1)	5	3
f(1,0)	1	2
f(0,0)	0	1

压栈情况如上图所示。当调用 f(7,91)时返回 7-1=6 程序结束。
答案为：n=6

例题 9：运行下列程序结果为_____。

```
#include <iostream>
void fun()
{
    for(int i=1;i<=3;i++)
    {
        static int var1=1;
        int var2=1;
        cout<<"var1="<<var1++
```

```
            <<",var2="<<var2++<<endl;
    }
}
void main()
{
    fun();
}
```

解答：
本题考查对静态局部变量的理解。静态局部变量存放在内存的全局数据区。函数结束时，静态局部变量不会消失，每次函数调用时，也不会为它重新分配空间，它始终驻留在全局数据区，直到程序运行结束。静态局部变量只在第一次调用时被初始化。在本程序中 for 循环 3 次，但是对静态变量 var1 的初始化只在第一次循环时完成，后两次循环不执行初始化。而变量 var2 在三次循环中都会被初始化。答案为：
var1=1,var2=1
var1=2,var2=1
var1=3,var2=1

例题 10：

```
#include <iostream>
void fun1(int);
void main()
{
    void fun1(double);
    fun1(1);

}
void fun1(int i)
{
    cout<<"int:"<<i<<endl;
}
void fun1(double i)
{
    cout<<"double:"<<i<<endl;
}
```

解答：
本题主要考查重载与作用域的关系问题。本程序中虽然定义了重载函数 fun1。但是在主函数中仅声明了参数类型为 double 的函数 fun1，然后就对 fun1 进行调用（调用前没有看到以整型数据作为形参的 fun1 函数的定义或声明）。那么尽管调用时赋的实参 1 是整型，也无法调用以整型数据作为形参的 fun1 函数。只能通过隐式数据类型转换与以 double 类型作为形参的 fun1 函数匹配。也就是说，在这里实际上没有实现重载。
所以，结果为：double:1。

例题 11：运行下列程序的结果为 var1=300,var2=400
var1=① ,var2= ② 。

```
#include <iostream>
void output(int var1,int var2)
{
    cout<<"var1="<<var1<<",var2="<<var2<<endl;
    var1=100;
    var2=200;
```

```
}
void main()
{
    int var1=300;
    int var2=400;
    output(var1,var2);
    cout<<"var1="<<var1<<",var2="<<var2<<endl;
}
```

解答：

本题主要考查变量的作用域问题。在主函数中定义的变量 var1 和 var2 其作用域为本程序范围，在函数 output 中形参 var1 和 var2 的作用域在 output 函数范围内，因为它和主函数中定义的变量同名，所以在该函数范围内会屏蔽主函数中定义的同名变量，其中的访问变量 var1 和 var2 的操作都是针对于形参变量，与主函数中定义的变量无关。当对 output 函数的调用结束后，形参变量消失，返回主函数，在主函数中访问的变量 var1 和 var2 仍为主函数中所定义的变量。所以，答案为：①300、②400

例题 12：写一个函数，它以一个名字作为命令行参数，打印输出“hello，name”（其中 name 为输入命令行参数）。

解答：

本程序主要考查对命令行参数的理解及使用。

说明：

```
void main(int argc,char* argv[]){…}
```

即是说，除按通常的那种无参方式来使用 main 函数外，如果需要的话，main 函数还可带有两个上述格式以及数据类型的形式参数。

main 带有形式参数时，其对应实参值由用户在执行该程序时指定，而后通过操作系统将它们传递给 main 函数。main 函数所含两个参数的含义如下：

argc-----第一参数，记录命令行参数的个数（其值为实际命令行参数的个数加 1）；

argv-----第二参数，为字符串数组，存放执行程序名以及各实际命令行参数。各数组元素的含义为：

argv[0]：本执行程序的文件名

argv[1]：第 1 个实际命令行参数（如果有）；

…

argv[argc-1]：第 argc-1 个实际命令行参数。

在 vc6.0 开发环境下设置命令行参数方法如下：

project→settings…→debug→”program arguments:”中，输入以空格分隔的各参数。

在命令提示符环境中，可通过如下形式运行程序：

程序文件名　参数 1 …

参数之间用空格隔开。

参考程序为：

```
#include <iostream>
void main(int argc,char* argv[])
{
    cout<<"hello,"<<argv[1]<<endl;
}
```

例题 13：使用重载函数的方法定义两个函数，用来分别求出两个 int 型数的点间距离和 double 型数的点间距离。两个 int 型点分别为 A(5,8),B(12,15)；两个 double 型点分别为 C(1.5,5.2),D(3.7,4.6)。

解答：

本题主要考查重载函数的应用。假设有两点 A(x1,y1),B(x2,y2)则两点之间距离为 $\sqrt{(x2-x1)^2+(y2-y1)^2}$ 。函数 sqrt()的作用是求平方根，使用该函数应包含头文件 math.h。这个程序中要实现求整型的两点之间距离和双精度型的两点之间距离，作用相同，但操作的对象类型不同，结果类型不同，所以用函数重载解决。
参考程序如下：

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

void main()
{
    double distance(int,int,int,int);
    double distance(double,double,double,double);
    int x1=5,y1=8,x2=12,y2=15;
    double xd1=1.5,yd1=5.2,xd2=3.7,yd2=4.6;
    double disi=distance(x1,y1,x2,y2);
    cout<<"两个 int 型数的点间距离："<<disi<<endl;
    double disd=distance(xd1,yd1,xd2,yd2);
    cout<<"两个 double 型数的点间距离："<<disd<<endl;
}
double distance(int a1,int b1,int a2,int b2)
{
    cout<<"\n 调用计算两个 int 型数的点间距离函数\n";
    double s=sqrt((a1-a2)*(a1-a2)+(b1-b2)*(b1-b2));
    return s;
}

double distance(double a1,double b1,double a2,double b2)
{
    cout<<"\n 调用计算两个 double 型数的点间距离函数\n";
    return sqrt((a1-a2)*(a1-a2)+(b1-b2)*(b1-b2));
}
```

例题 14：使用递归函数求解并输出 fibonacci 数列的前十项。
解答：
本题主要考查对递归函数的应用。对于 fibonacci 数列在前边已经介绍，这里使用递归完成，请读者对比递归于非递归编程的方法。
参考程序如下：

```
#include <iostream>
#include <iomanip>
int fib(int n)
{
    if(n<3) return 1;
    return (fib(n-2)+fib(n-1));
}
void main()
{
    for(int i=1;i<=10;i++)
        cout<<setw(5)<<fib(i);
```

```
}
```

例题 15：打印某一年的年历。
解答：
关于本题的编程细节请参见程序中的注释。
参考程序如下：

```
#include <iostream>
#include <iomanip>

int FirstDayOfYear(int y);
int DaysOfMonth(int m);
void PrintMonth(int m);
void PrintHead(int m);
bool IsLeapYear(int y);

int weekDay;
int year;

void main()
{
    cerr<<"请输入您想要打印的年份:\n";
    cin>>year;
    if(year<1900)
    {
        cerr<<"年份不能小于 1900。\n";
        return;
    }
    weekDay=FirstDayOfYear(year);//一年的第一天星期几

    //打印年标题
    cout<<"\n\n\n\n\n                              "<<year<<" 年日历\n";
    cout<<"\n  ================================================";

    //打印每个月
    for(int i=1;i<=12;i++)
        PrintMonth(i);
    cout<<endl;
}

//某个月打印函数
void PrintMonth(int m)
{
    PrintHead(m); //打印月标题
    int days=DaysOfMonth(m);  //该月的天数
    for(int i=1;i<=days;i++)
    {
        cout<<setw(6)<<i;
        weekDay=(weekDay+1)%7;
        if(weekDay==0)  //打印下一天时判断是否换行
        {
```

```
            cout<<endl;
            cout<<"          ";//行首空格
        }
    }
}

//打印月标题
void PrintHead(int m)
{
    cout<<"\n\n"<<setw(6)<<m<<"月      日    一    二    三    四    五    六
\n";
    cout<<"          ";
    for(int i=0;i<weekDay;i++)
        cout<<"      ";
}

//判断某月天数的函数
int DaysOfMonth(int m)
{
    switch(m){
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:return 31;
        case 4:
        case 6:
        case 9:
        case 11:return 30;
        case 2:if(IsLeapYear(year))
                    return 29;
               else
                    return 28;
    }
    return 0;
}

//判断是否为闰年
bool IsLeapYear(int y)
{
    return((y%4==0&&y%100!=0)||year%400==0);
}

//判断某年的第一天
//因为每年都是52个整星期多一天，因此n年多出的天数还是n天，
//再加上1900年到year年间因闰年多出来的天数，
//再加上1900年元旦的星期序号1与7取模便得出是第几天。由于2000年正好是闰年，所
//以可以计算到2099年。
int FirstDayOfYear(int y)
```

```
{
    int n=y-1900;
    n=n+(n-1)/4+1;
    n=n%7;
    return n;
}
```

【习题】

一、选择题

1.下列函数定义语句正确的是（　）。

(a)

```
void fun(int var1)
{
    int var1=0;
    cout<<var1<<endl;
}
```

(b)

```
void fun(int var1,var2)
{
    cout<<var1+var2<<endl;
}
```

(c)

```
int fun(int var1)
{
    if(var1)
        return 1;
    else
        return 0;
}
```

(d)

```
int fun(int var1)
{
    if(var1)
        return 1;
    else
        cout<<0<<endl;
}
```

2.下列叙述中正确的是(　)

(a)C++语言程序中，main()函数必须在其它函数之前，函数内可以嵌套定义函数。

(b)C++语言程序中，main()函数的位置没有限制，函数内不可以嵌套定义函数。

(c)C++语言程序中，main()函数必须在其它函数之前，函数内不可以嵌套定义函数。

(d)C++语言程序中，main()函数必须在其它函数之后，函数内可以嵌套调用函数。

3.下列对 return 语句叙述错误的是（　）。

(a)在函数定义中可能有 return 语句，也可能没有 return 语句。

(b)在函数定义中可以有多条 return 语句。

(c)在函数定义中每条 return 语句可能返回多个值。

(d)如果函数类型不是 void 型，则函数定义中必须有 return 语句。

4.C++语言中函数返回值的类型是由（　）决定的。
(a)return 语句中的表达式类型　　(b)调用该函数的主调函数类型
(c)定义函数时所指定的函数类型　　(d)以上说法都不正确

5.C++中，关于参数默认值的描述正确的是（　）。
(a)只能在函数定义时设置参数默认值
(b)设置参数默认值时，应当从右向左设置
(c)设置参数默认值时，应当全部设置
(d)设置参数默认值后，调用函数不能再对参数赋值

6.使用重载函数编程序的目的是（　）。
(a)使用相同的函数名调用功能相似但参数不同的函数　　(b)共享程序代码
(c)提高程序的运行速度　　(d)节省存储空间

7.系统在调用重载函数时，下列不能作为确定调用哪个重载函数的依据的选项是（　）。
(a)函数名　　(b)参数个数　　(c)函数类型　　(d)参数类型

8.运行下列程序结果为（　）。
```
#include <iostream>
inline int abs(int x)
{
 return x<0?-x:x;
}
void main()
{
 int var1=10,var2=-10;
 cout<<" |var1|+|var2|="<<abs(var1)+abs(var2)<<endl;
}
```
(a) |var1|+|var2|=0　　(b) |var1|+|var2|=20
(c) |var1|+|var2|=10　　(d) |var1|+|var2|=-10

9.数组作为函数的形参，把数组名作为函数的实参时，传递给函数的是（　）。
(a)数组中各元素的值
(b)数组中元素的个数
(c)数组中第 0 个元素的值
(d)该数组的首地址

10.运行下列程序结果为（　）。
```
#include  <iostream>
int f(int[][3],int,int);
void main()
{
    int a[][3]={0,1,2,3,4,5,6,7,8};
    cout<<f(a,3,3)<<endl;
}
int f(int a[][3],int row,int col)
{

    int i,j,t=1;
    for(i=0;i<row;i++)
```

```
        for(j=0;j<col;j++)
            {a[i][j]++;
             if(i==j) t*=a[i][j];
             }
    return t;
}
```

(a)0　　(b)48　　(c)105　　(d)45

11.运行下列程序的输出结果为（　）。

```
#include <iostream>
int var;
main()
{
   int var=2;
   ::var=0;
   if(var>1)
   {
     int var=5;
     cout<<var;
    }
    cout<<var;
    cout<<::var<<endl;
    return 0;
}
```

(a)20　　(b)000　　(c)520　　(d)500

二、填空题

1.一个 C++语言程序总是从_____开始执行。

2._____使编译器可以检查传入函数的参数个数、类型和顺序。

3.在函数原型说明中必须包含的要素有函数类型（如果省略则默认为 int 型）、_____、_____。

4._____限定符声明只读变量。

5.若某个函数没有返回值，则该函数的类型应定义为_____类型。

6.一个函数直接或间接地调用自身，这种现象称为函数的_____。

7.在一个函数的定义或声明前加上关键字_____则就把该函数定义为内联函数，它主要是解决_____问题。

8.函数的参数传递的方式分为两类，分别是_____方式和_____方式。

9.在 c++中，可以有多个同名而处理不同参数类型或个数的函数，称为函数_____。

10.标识符的四种作用域是_____、_____、_____、_____。

11.要让函数中的局部变量在函数调用之间保持其数值，则要用存储类说明符_____声明。

12.运行下列程序的结果为_____。

```
#include  <iostream>
void swap(int &,int &);
void main()
{
 int a=66,b=4;
 cout<<"a="<<a<<",b=“
     <<b<<endl;
 swap(a,b);
 cout<<"a="<<a<<",b=“
     <<b<<endl;
```

```
}
void swap(int &x,int &y)
{
 int t=x;
 x=y;
 y=t;
}
```

13.运行下列程序的结果为_____。

```
#include  <iostream>
#include  <iomanip>
void fun(int array[ ],int n);
void main(  )
{
    int a[10]={1,1};
    int i;
    fun(a,10);
    for(i=0;i<10;i++)
           cout<<setw(4)<<a[i];
    cout<<endl;
}
void fun(int array[ ],int n)
{
    int i;
    for(i=2;i<n;i++)
          array[i]=array[i-1]+array[i-2];
}
```

14.运行下列程序，若输入 1 2 3 4 5 则输出结果为_____。

```
#include <iostream>
void rev(int n)
{
    int x;
    cin>>x;
    if(n==1)
        cout<<x;
    else
    {
        rev(n-1);
        cout<<x;
    }
}
void main()
{
    rev(5);
}
```

15. 运行下列程序结果为_____。

```
#include <iostream>
void fun(char PrChar='$',int num=10);
main()
{
```

```
  char ch;
  int num;
  ch='#';
  num=20;
  fun(ch,num);
  fun();
  fun('&');
  return 0;
}
void fun(char ch,int num)
{
   for(int i=0;i<num;i++)
   {
     cout<<ch;
   }
    cout<<"\n";
}
```

16.运行下列程序结果为_____。

```
 #include  <iostream>
void fun( );
void main(  )
{
    int i;
    for(i=0;i<5;i++)
           fun();
}
void fun(  )
{
    static int m=0;
    cout<<m++;
}
```

三、编程题

1. 编写函数将华氏温度转换为摄氏温度，公式为$c=(F-32)\times 5\div 9$；并在主函数中调用。
2. 编写函数利用全局变量统计数组中奇数和偶数的个数。
3. 利用重载编写求整数绝对值和求实数绝对值两个函数。
4. 编写函数利用递归的方法计算 x 的 n 阶勒让德多项式的值。该公式如下：

$$Px(x)=\begin{cases}1 & (n=0)\\ x & (n=1)\\ ((2*n-1)*x*P_{n-1}(x)-(n-1)*P_{n-2}(x))/n & (n>1)\end{cases}$$

5..编写程序，验证哥德巴赫猜想：任何大于 2 的偶数都是两个素数之和（在 1000 以内验证）。

【参考答案】

一、选择题

1-5. c b c c b 6-10. a c b d d 11. c

二、填空题

1. main 函数
2. 函数原型
3. 函数名、参数表
4. const
5. void
6. 递归调用
7. inline、程序的运行效率
8. 值传递、引用传递（或地址传递）
9. 重载
10. 函数范围、文件范围、块范围、函数原型范围
11. static
12.
a=66,b=4
a=4,b=66
13. 1 1 2 3 5 8 13 21 34 55
14. 54321
15.
##########
$$$$$$$$$$
&&&&&&&&&&
16.01234
三、编程题
1.
```
#include <iostream>
double ftoc(double);
void main()
{
	cout<<"华氏温度"<<104.0<<"度为摄氏"<<ftoc(104.0)<<"度！"<<endl;
}
double ftoc(double f)
{
	double c;
	c=(f-32)*5/9;
	return c;
}
```

2.
```
#include <iostream>
int numo=0,nume=0;
void sta(int a[],int);
void main()
{
	const int N=10;
	int array[N]={0,2,3,4,5,6,7,8,10,12};
	sta(array,N);
	cout<<"奇数有"<<numo<<"个,"
		<<"偶数有"<<nume<<"个"<<endl;
}
void sta(int a[],int n)
{
	for(int i=0;i<n;i++)
		if(a[i]%2==0)
```

```
                nume++;
            else
                numo++;
}
3.
#include <iostream>
int abs(int);
double abs(double);
void main()
{
    int a=-8;
    double b=-3.14;
    cout<<abs(a)<<","
        <<abs(b)<<endl;
}
int abs(int x)
{
    return (x>=0?x:-x);
}
double abs(double x)
{
    return (x>=0?x:-x);
}
4.
#include <iostream>
double p(double,int);
void main()
{
    cout<<38.14<<"的"<<5<<"阶勒让德多项式的值为："<<p(38.14,5)<<endl;

}
double p(double x,int n)
{
    if(n==0)
        return 1;
    else
        if(n==1)
            return x;
        else
            return ((2*n-1)*x*p(x,n-1)-(n-1)*p(x,n-2))/n;
}
5.
#include <iostream>
#include<math.h>
int isprime(int);
void main()
{
    cout<<4<<"="<<2<<"+"<<2<<endl;
    for(int i=6;i<=100;i+=2)
        for(int j=3;j<=i/2;j+=2)
        {
            if(isprime(j)&&isprime(i-j))
            {
                cout<<i<<"="<<j<<"+"<<(i-j)<<endl;
```

```
                        break;
                }
            }
}
int isprime(int n)
{
    int x=sqrt(n);
    for(int i=2;i<=x;i++)
        if(n%i==0)
            return 0;
    return 1;
}
```

第 6 章　指针、引用和动态空间管理

【典型例题】

例题 1：下列语句正确的是（　）。
(a)int *p=0;
(b)int var1=100,*p=var1; //cannot convert from 'int' to 'int *'
(c)int var1=100;char *p=&var1;// cannot convert from 'int *' to 'char *'
(d)int var1=100,*q;void *p=&var1;q=p;// cannot convert from 'void *' to 'int *'
解答：
本题主要考查对指针的理解。指针不能被赋予非地址值，指针不能被初始化或赋值为其他类型对象的地址值。选项 a 中 p 被初始化为没有指向任何对象，这是可以的；选项 b 中定义了一个整型变量和一个指向整型的指针，不能把一个整型变量赋值给一个指向整型的指针，指针用来存放变量的地址，所以选项 b 是错误的；选项 c 中将一个整型变量的地址赋给了指向字符型量的指针，这是错误的；选项 d 中将一个整型变量的地址赋给 void 型指针是错误的。所以答案为：a。

例题 2：有如下定义　int array[3]={0,1,2},*p=array;，则要访问数组中的第二个元素，下列语句错误的是（　）。
(a) *(array+1)　　(b) array[1]　　(c) *array + 1　　(d) *(p+1)
解答：
数组名 array 和指针变量 p 都存放数组的首地址，所以选项 a 和 d 都是求出第二个元素的地址，然后取该地址中的值，所以这两个选项实现了对数组中第二个元素的访问；选项 b 为下标法访问数组元素；选项 c 的作用是求出数组中第一个元素的值，然后用这个值加 1。所以，本题答案为：c。

例题 3：程序改错题

```
#include <iostream>
#include<string.h>
void main()
{
    const char *p="hello world!"; //①
    int len=0;
    while(p++) len++;//②
    p-=len+1;
    cout<<"the lenth of "<<p<<" :"<<len<<endl;//③
}
```

答案为：while(*p++) len++;
解答：
本题主要考查对地址和地址中的值得区分和应用。这个程序的作用是求字符串的长度。这里计算长度的循环语句中的循环条件是只要当前字符不为'\0'表示字符串还没有结束，则长度加 1，指针移到字符串中的下一个字符。所以程序中语句②错误，它没有判断字符而是判断地址是否存在，这是错误的。应改为：while(*p++) len++;

例题 4：对于定义 int *f()中，标识符 f 代表的是（　）。
(a)一个指向函数的指针　　(b)一个指针型函数，该函数返回值为指针
(c)一个指向整型数据的指针　　(d)一个指向数组的指针
解答：

本题主要考查对指针函数和函数指针的理解。这里定义的是指针型函数，也就是说这个函数的返回值是指针。所以答案为：b。

例题 5：运行下列程序结果为_____。

```
#include <iostream>
#include <cstdlib>
const int N=3;
void process(float *p,int n,float (*fun)(float *p,int n));
float arr_add(float arr[],int n)
{
    int i;
    float sum=0;
    for(i=0;i<n;i++)
        sum=sum+arr[i];
    return(sum);
}
float odd_add(float *p,int n)
{
    int i;
    float sum=0;
    for(i=0;i<n;i=i+2,p=p+2)
        sum=sum+*p;
    return(sum);
}
float arr_ave(float *p,int n)
{
    int i;
    float sum=0,ave;
    for(i=0;i<n;i++)
      ave=sum/n;
    return(ave);
}
float arr_max(float arr[],int n)
{
    int i;
    float max;
    max=arr[0];
    for(i=0;i<n;i++)
        if(arr[i]>max)
        max=arr[i];
    return(max);
}
void process(float *p,int n,float (*fun)(float *p,int n))
{
    float result;
    result=(*fun)(p,n);
    cout<<result;
}
void main()
{
void process(float *p,int n,float (*fun)(float *p,int n));
    float arr_add(float arr[],int n);
    float odd_add(float *p,int n);
    float arr_ave(float *p,int n);
    float arr_max(float arr[],int n);
    static float a[]={1.5,3.8,5.6};
    int n=N;
```

```
    cout<<"\nthe sum of "<<n<<"element is :"<<"\n";
    process(a,n,arr_add);
    cout<<"\nthe sum of old element is :"<<"\n";
    process(a,n,odd_add);
    cout<<"\nthe sum of average of"<<n<<" element is :"<<"\n";
    process(a,n,arr_ave);
    cout<<"\nthe sum of maximum of"<<n<<" element is :"<<"\n";
    process(a,n,arr_max);
}
```

解答：

本题主要考查对函数指针的理解。本程序的作用是通过调用一系列的函数完成一些运算。函数 arr_add()，odd_add()，arr_ave()，arr_max()的作用分别是求数组元素的和，求奇数位元素的和（即求下标为偶数的元素的和），求数组中元素的平均值，求数组中元素的最大值。函数 process(float *p,int n,float (*fun)(float *p,int n))的作用是通过函数指针调用不同的函数（其中第三个参数为函数指针）。在主函数中调用 process()函数，传递的第一个实参为数组首地址，第二个实参位数组长度，第三个实参为函数名，用于传递函数的首地址。答案为：

```
the sum of 3element is:
10.9
the sum of old element is:
7.1
the sum of average of3 element is:
0
the sum of maximum of3 element is:
5.6
```

例题 6：对于下列函数定义，说法正确的是（　）。

```
void fun1()
{
    int var1=2,var2=3;
    int *p=&var1,*q=&var2;
    p=q;

}
void fun2()
{
    int var1=2,var2=3;
    int &p=var1,&q=var2;
    p=q;
}
```

(a) fun1 与 fun2 的作用完全相同。

(b) 运行 fun1 后 p、q 中均存放 var2 的地址；运行 fun2 后 p、q 均为 var2 的别名。

(c) 运行 fun1 后 p 中存放 var1 的地址，q 中存放 var2 的地址；运行 fun2 后 p 为 var1 的别名，q 为 var2 的别名。

(d) 运行 fun1 后 p、q 中均存放 var2 的地址；运行 fun2 后 p 为 var1 的别名，q 为 var2 的别名。

解答：

本题主要考查引用和指针的区别。对于函数 fun1()，p 和 q 为指向整型的指针，它们分别被初始化为 var1 和 var2 的地址，p=q 则使 p 中存放的地址为 q 中的地址即为 var2 的地址，所以最终 p,q 里都存放 var2 的地址。对于函数 fun2()，p 和 q 为引用，它们分别被初始化为 var1 和 var2 的别名，引用一旦被初始化，它就不能在指向其他对象。p=q 的作用是让 var1 的值等于 var2 的值，即这个语句执行以后 var1 和 var2 的值都为 3，而 p 和 q 仍然分别为 var1 和 var2 的别名。引用必须被初始化为指向一个对象，一旦初始化了它就不能再指向其他对象；指针可以指向一系列不同的对象也可以什么都不指向；如果一个参数可能在函数中指向不同的对象或者这个参数可能不指向任何对象则必须使用指针参数。答案为：d

例题 7：编写一个程序完成字符串比较。当两个字符串长度相等时，对其中各个字符进行比较，比到第一个不相等的字符为止，字符大的该字符串就大；若两个字符串长度不等，则长度小的字符串小；当两个字符串长度相等，对应位字符也相等时，两个字符串相等。

解答：

参考程序为；

```
#include <iostream>
using namespace std;
int strcmp1(const char *str1,const char *str2);
void main()
{
    char s1[100],s2[100];
    int result;
    cout<<"请输入两个字符串：";
    cin>>s1>>s2;
    result=strcmp1(s1,s2);
    if(result>0)
        cout<<s1<<">"<<s2<<endl;
    else
        if(result<0)
            cout<<s1<<"<"<<s2<<endl;
        else
            cout<<s1<<"="<<s2<<endl;
}
int strcmp1(const char *str1,const char *str2)
{
    while(*str1!='\0'&&*str2!='\0')
    {
        if(*str1==*str2)
        {
            str1++;str2++;
        }
        else
            return *str1-*str2;
    }
    return *str1-*str2;
}
```

例题 8：输入 10 个国家名称，用指针数组实现排序输出。

解答：

本题主要考查对指针数组的应用。这里用指针数组存放十个国家，并将指针数组作为参数。则数组中的每个元素都是一个指针用于存放字符串的首地址。通过字符串比较函数，比较字符串大小。排序可以采用任何一种排序方法，这里用简单选择排序。

参考程序为：

```
#include <iostream>
#include<string.h>
using namespace std;
void ccmp(char *a[]);
void main()
{
    char *cname[10]={"Afghanistan","Australia","Brazil",
                     "Oman ","Romania ","Singapore","Zambia",
                     "Spain ","Mexico","Canada"};
    ccmp(cname);
    for(int i=0;i<10;i++)
        cout<<cname[i]<<endl;
}
```

```
void ccmp(char *a[10])
{
    char *p;int i,j;
    for(i=0;i<9;i++)
        for(j=i+1;j<10;j++)
        {
            if(strcmp(a[i],a[j])>0)
            {
                p=a[i];
                a[i]=a[j];
                a[j]=p;
            }
        }
}
```

【习题】

一、选择题

1. 要使变量 i 成为 int 型变量 x 的别名，正确的定义语句是（ ）。

(a) int &i=x; (b) int i=&x; (c) int &i=&x; (d) int i=x;

2. 在下列指针表达式中，与下标访问 a[i][j]不等效的是（ ）。

(a) *(a+i+j) (b) (*(a+i))[j] (c) *(*(a+i)+j) (d) *(a[i]+j)

3. 已定义字符串 char str[5],则下列表达式中不能表示 str[1]的地址的是（ ）。

(a) str+1 (b) str++ (c) &str[0]+1 (d)&str[1]

4. 已知：int a[]={1,2,3,4,5,6},*p=a,x；下面语句中 x 的值为 5 的是（ ）。

(a) p+=3;x=*(p++); (b) p+=5;x=*p++ ; (c) p+=4;x=*++p; (d) p+=4; x=*p++

5. 若有说明：int i,j=6,*p;p=&i；则与 i=j 等价的语句是（ ）。

(a) i=*p; (b) *p=*&j; (c) i=&j; (d)i=**p;

6. 设 p1 和 p2 是指向同一个 int 型一维数组的指针变量，k 为 int 型变量，则不能正确执行的语句是（ ）。

(a) k=*p1+*p2; (b) p2=k; (c) p1=p2; (d) k=*p1*(*p2);

7. 下面函数的功能是（ ）。

```
int fun(char *x)
  {
    char *y=x;
    while(*y++){};
    return y-x-1;
    }
```

(a) 求字符串的长度 (b) 求字符串存放位置

(c) 比较两个字符串的大小 (d) 将字符串 x 连接到字符串 y 后面

8. 执行以下程序段后，m 的值为（ ）。

```
int a[2][3]={{1,2,3},{4,5,6}};
int m,*p=&a[0][0];
m=(*p)*(*(p+2))*(*(p+4));
```

(a) 15 (b) 14 (c) 13 (d) 12

9. 设有如下定义，下面关于 ptr 正确叙述是（ ）。

```
int (*ptr)();
```

(a) ptr 是指向一维数组的指针变量。

(b) ptr 是指向 int 型数据的指针变量。

(c) ptr 是指向函数的指针，该函数返回一个 int 型数据。
(d) ptr 是一个函数名，该函数的返回值是指向 int 型数据的指针。

10. 若有如下语句：

```
int **pp,*p,a=10,b=20;
pp=&p;
p=&a;
p=&b;
cout<<*p<<",,"<<**pp<<endl;
```

则输出结果是（ ）。

(a) 10,20 (b) 10,10 (c) 20,10 (d) 20,20

二、填空题

1. 运行下列程序结果为_____。

```
#include   <iostream>
void main()
{ int a[3]={10,15,20};
int *p1=a,*p2=&a[1];
*p1=*(p2-1)+5;
*(p1+1)=*p1-5;
cout<<a[1]<<endl;
}
```

2. 运行下列程序结果为 max=15,min-=-5，请将程序补充完整。

```
#include <iostream>
using namespace std;
const int N=10;
void max_min(int arr[],int *pt1,int *pt2,int n)
{
    int i;
    *pt1=*pt2=arr[0];
    for(i=1;i<n;i++)
    {
        if(arr[i]>*pt1) _____①_____
        if(arr[i]<*pt2) _____②_____
    }
    return ;
}
void main()
{
    void max_min(int arr[],int *pt1,int *pt2,int n);
    int   array[N]={1,8,10,2,-5,0,7,15,4,-5};
    int *pt1,*pt2,a,b;
    pt1=&a;
    pt2=&b;
    max_min(_____③_____);
    cout<<"max="<<a<<",min="<<b<<"\n";
}
```

3. 运行下列程序结果为_____。

```
#include <iostream>
#include<stdlib.h>
using namespace std;
char * month_name(int n)
{
```

```
    static char *name[]={
        "ILLEGAL MONTH","JANUARY",
        "FEBRUARY","MARCH","APRIL",
        "MAY","JUNE","JULY","AUGUST",
        "SEPTEMBER","OCTOBER","NOVEMBER","DECEMBER"};
        return((n<1||n>12)?name[0]:name[n]);
}
void main()
{
    char * month_name(int n);
    cout<<"MONTH NO"<<3<<"    "<<month_name(3)<<"\n";
}
```

4. 运行下列程序结果为_____。

```
#include <iostream>
using namespace std;
void callbyval(int a,int b,int c)
{
a=3;b=2;c=1;
}
void callbypointer(int* a,int* b,int* c)
{
*a=3;*b=2;*c=1;
}
void callbyreference(int& a,int& b,int& c)
{
a=1;b=2;c=3;
}
void main()
{
int a=1,b=2,c=3;
int& a1=a;
int& b1=a;
int& c1=a;
callbyval(a,b,c);
cout<<a<<b<<c<<endl;
callbypointer(&a,&b,&c);
cout<<a<<b<<c<<endl;
callbyreference(a1,b1,c1);
cout<<a<<b<<c<<endl;
}
```

三、编程题

1.在很多场合填写金额时，为了防止篡改，要求同时填写数字和大写汉字金额。编写一个程序，输入数字金额，输出其汉字金额（该程序仅处理金额为 0.00-99999.99 元）。如金额 10234.56 元，写成壹万零贰百叁拾肆元伍角陆分。

2.使用引用参数编制程序，实现两个字符串变量的交换。例如开始时：

char *ap=”hello”；

char *bp=”how are you”；

交换后使 ap 和 bp 指向的内容别是：

ap：”how are you”

bp：”hello”

3. 编程建立一个有序单链表，对该链表完成如下操作：

①输出该链表

②求表长
③插入一个元素，保持有序性
④删除链表中第 i 个位置上的数

【参考答案】

一、选择题
1-5、a a b d b　　2-10.、b a a cd

二、填空题
1. 10
2. ①*pt1=arr[i];②*pt2=arr[i];③array,pt1,pt2,N
3. MONTH NO3 MARCH
4. 123
321
321

三、编程题
1.
参考程序为：

```
#include <iostream>
#include <stdlib.h>
using namespace std;
void main()
{
	double n;
	const char* a[10]={"零","壹","贰","叁","肆","伍","陆","柒","捌","玖"};
	const char* c[8]={"万","仟","佰","拾","元","角","分"};
	int b[7]={0};
	int i,j,m,k;

	cout<<"\n 请输入金额(0.00~99999.99)：\n";
	cin>>n;
	if(n<0||n>99999.99)
	{
		cout<<"输入有误！\n";
		exit(0);
	}
	if(n==0)
		cout<<"汉字大写为：零元\n";
	else
	{
	k=n*100;
	i=6;
	while(k!=0)
	{
		m=k%10;
		b[i]=m;
		k/=10;
		--i;
	}

	cout<<"汉字大写金额为：\n";
```

```
        for(j=0;j<=6;j++)
        {
            if(b[j]!=0)
                break;
        }
        for(k=6;k>=0;k--)
        {
            if(b[k]!=0)
                break;
        }
        for(i=0;i<=6;i++)
        {
            m=b[i];
            if(b[i+1]!=0&&b[i]==0&&i<=3&&i>j&&i<=k)
                cout<<a[0];
            if(m!=0)
                cout<<a[m]<<c[i];
            else
                if(i==4&&n>=1)
                    cout<<c[i];
        }
            cout<<endl;
}
}
```

2.

参考程序为：

```
#include    <iostream>
void Swap(char*& str1, char*& str2);

void main()
{
   char* ap="hello";
   char* bp="how are you?";
   cout <<ap <<endl <<bp <<endl;

   Swap(ap, bp);

   cout <<"交换以后:\n";
   cout <<ap <<endl <<bp <<endl;
}

void Swap(char*& str1, char*& str2)
{
   char* temp=str1;
   str1=str2;
   str2=temp;
}
```

3.

参考程序为：

```
#include <iostream>
#include <malloc.h>
using namespace std;
```

```
struct node
{
    int data;
    node *next;
};

int creat(node* h)      //逆序建立线性链表
{
    node *p,*s;
    int a;
    p=h;
    cout<<" 请按从大到小的顺序输入结点值,输入 0 结束：\n";
    cout<<"请输入结点的值";
    cin>>a;
    while(a!=0)
    {
    if(!(s=(node *)malloc(sizeof(node))))    //分配结点空间，可以该用 new 操作
            return 1;
        else
        {
            s->data=a;
            s->next=p->next;          //每次将新建结点 s 插入到头结点后（逆序）
            p->next=s;
            cout<<"请输入结点值:";
            cin>>a;
        }
    }
    return 0;
}

void showlist(node* h)   //顺序显示链表内容
{
    node *q;
    q=h->next;
    cout<<"链表内容如下："<<endl;
    while(q!=NULL)        //遍历
    {
        cout<<"->"<<q->data;
        q=q->next;
    }
    cout<<endl<<endl<<endl;
}

int len(node *h)    //求表长（不包含头结点）
{
    node *p=h->next;
    int i=0;
    while(p)
    {
```

```
            i++;
            p=p->next;
        }
        return i;
}

int insert(node *h)     //插入一个元素，保持有序性
{
        int a;
        node *s,*p=h;
        cout<<"请输入要插入的数 \n";
        cin>>a;
        if(!(s=(node*)malloc(sizeof(node))))
            return 1;
        else
        {
            s->data=a;
            while(p->next&&p->next->data<a)
            {
                p=p->next;
            }
            s->next=p->next;
            p->next=s;
        }
        return 0;
}
int del(node *h)   //删除链表中第 i 个位置上的数
{
        int i,j;
        node *p=h,*q;
        cout<<"  请输入要删除数的位置：";
         cin>>i;
        if(i<=0||i>len(h))
        {
            cout<<"位置有误!\n";
            return 1;
        }
        else
        {
            for(j=1;j<i;j++)
            {
                p=p->next;     //p 记录待删除结点的前一结点
            }
            q=p->next;   //q 记录待删除结点
            p->next=q->next;
            free(q);
        }
        return 1;
}
```

```
void main()
{
	node *head;

	if(!(head=(node*)malloc(sizeof(node))))
		return;
	else
	{
	head->next=NULL;
	 creat(head);
	showlist(head);
	cout<<"链表长度为 ： "<<len(head)<<endl<<endl;
	 insert(head);
	showlist(head);
	del(head);
	showlist(head);
	}
}
```

第 7 章　类和对象的创建

【典型例题】

例题 1．下列程序段是否有错，若有错请改错。

```
#include <iostream>
using namespace std;
class point----------------------------------------------①
{
      private:
            int x,y;
      public:
            void setpoint(int, int);--------------------②
};
int point:: setpoint(int xx, int yy)   ----------------③
{
      x=xx;---------------------------------------------  ④
      y=yy;---------------------------------------------  ⑤
      return 1;-----------------------------------------  ⑥
}
void main()
{
      point p1;-------------------------------------------------------------------------  ⑦
      p1.setpoint(2,4);-----------------------------------------------------------------  ⑧
      cout<<"坐标为：("<<p1.x<<","<<p1.y<<")"<<endl;-------------------------  ⑨
}
```

解答：

这里⑨错误，不能在类定义以外直接访问类的私有成员。要得该点的两个坐标，应该在类中定义获取私有成员 x, y 的公有成员函数，使得可以在类外通过类的公有成员函数对其私有成员进行间接访问。所以程序应改为：

```
class point
{
      private:
            int x,y;
      public:
            void setpoint(int, int);    //声明成员函数 setpoint()的函数原型
            int getx();       //声明成员函数 getx()的函数原型
            int gety();       //声明成员函数 gety 的函数原型
};
void point:: setpoint(int xx, int yy) //定义成员函数 setpoint()
{
      x=xx;
      y=yy;
}
int point :: getx()   //定义成员函数 getx()
{     return x;}
int point :: gety()   //定义成员函数 gety
{     return y;}
```

例题 2．在下列函数原型中，可以作为类 student 的构造函数的说明的是（　）。
(a)void student(int age);　　(b)int student( );　　(c)student(int)const;　　(d)student(int);
解答：
本题主要考查对构造函数的特点的掌握。构造函数的名字必须与类的名字相同。构造函数没有返回值，不能定义返回类型，包括 void 型在内。构造函数可以是内联函数，可带有参数表，可带有默认的形参值，还可重载。选项 a、b 均有返回值类型，不能作为构造函数。选项 c 为常成员函数，构造函数不能为常成员函数。答案为：d

例题 3．下列说法正确的是（　）。
(a) 可以定义修改对象数据成员的 const 成员函数。
(b) 不允许任何成员函数调用 const 对象，除非该成员函数也声明为 const。
(c) const 对象可以调用非 const 成员函数。
(d) const 成员函数可以调用本类的非 const 成员函数。
解答：
c++编译器不允许任何成员函数调用 const 对象，除非该成员函数本身也声明为 const。声明 const 的成员函数不能修改对象，因为编译器不允许其修改对象。对 const 对象调用非 const 成员函数是个语法错误。定义调用同一类实例的非 const 成员函数的 const 成员函数是个语法错误。答案为：b

例题 4．运行下列程序后，” constructing A!” 和” destructing A!”分别输出几次（　）。

```
#include <iostream>
using namespace std;
class A
{
    int x;
public:
    A()
    {cout<<" constructing A!"<<endl;}
    ~A()
    {cout<<" "<<endl;}
};
void main()
{
    A a[2];
    A *p=new A;
    delete p;
}
```

(a)2 次，2 次　　(b)3 次，3 次　　(c)1 次，3 次　　(d)3 次，1 次
解答：
本题主要考查在什么情况下系统会调用构造函数与析构函数。在主函数中定义了一个对象数组，其中有两个元素，该数组中的每个元素都是一个类的对象，所以这里会调用 2 次构造函数；new A 时创建一个 A 类的对象，所以也会调用构造函数，因此一共调用 3 次构造函数。delete p;会撤消 new 运算分配的空间，它会调用 1 次析构函数。主函数结束时要释放数组所占空间，会调用 2 次析构函数，因此析构函数也调用了 3 次。答案为：b

例题 5．运行下列程序的结果为_____。

```
#include <iostream>
#include<string.h>
using namespace std;
class course
{
    int id;
    char name[50];
public:
```

```
    course(int csid,char *csname)
    {
        cout<<"constructing course!"<<endl;
        id=csid;
        strcpy(name,csname);
    }
    ~course()
    {
        cout<<"destructing course!"<<endl;
    }
    int getid()
    {return id;}
    char* getname()
    {
        return name;
    }
};
class student
{
    char name[10];
    int age;
    course c1;
public:
    student(char *sname,int sage,int cid,char *cname):c1(cid,cname)
    {
        cout<<"constructing student!"<<endl;
        strcpy(name,sname);
        age=sage;
    }
    ~student()
    {
        cout<<"destructing student!"<<endl;
    }
    void prints()
    {
        cout<<"name:"<<name<<endl
            <<"age:"<<age<<endl
            <<"course id:"<<c1.getid()<<endl
            <<"course name:"<<c1.getname()<<endl;
    }
};

void main()
{
    student st1("tom",23,1,"c++ programming language");
    st1.prints();
}
```

解答：

本题主要考查在为含有对象成员的类创建对象时，构造函数的调用顺序，对象成员的初始化问题以及对象撤消时调用析构函数的顺序。对于本程序，在主函数中创建 student 类的对象则调用其构造函数 student()，该构造启动时，首先为数据成员分配空间，然后根据在类中声明的对象成员的顺序依次调用其构造函数，在这里调用 course 类的构造函数，最后才执行自己的构造函数的函数体。析构函数以调用构造函数相反的顺序被调用。

答案为：

constructing course!
constructing student!

```
name:tom
age:23
course id:1
course name:c++ programming language
destructing student!
destructing course!
```

例题 6．运行下列程序输出结果为_____。

```
#include <iostream>
using namespace std;
class A{
public:
    A(int X){cout<<"ok!";}
    A(){}
};
int main()
{
    A a[3],a1(3);
    return 0;
}
```

解答：

本题主要考查对重载构造函数的理解。这里创建对象数组时，对数组的每一个元素都将调用一次构造函数，如果没有显式给出数组元素的初值，则调用缺省构造函数。而创建对象 a1 时带有一个整型参数，所以调用以整型作为参数的构造函数，它输出 ok！。所以，本题答案为：ok！

例题 7．运行下列程序结果为_____。

```
#include <iostream>
const double PI=3.14159;
class circle
{
    double r;

public:
    static int num;
    circle(double);
    circle(circle &);
    double getr();
};
circle::circle(double i)
{
    r=i;
}
circle::circle(circle &c)
{
    num++;
    cout<<"第"<<num<<"次调用拷贝构造函数！"<<endl;
    r=c.r*num;
}
double circle::getr()
{
    return r;
}
double getradius(circle c3)
{
    return c3.getr();
```

```
}
circle fun1()
{
    circle c4(5);return c4;
}
int circle::num=0;
void main()
{
    circle c1(1);
    cout<<"c1:"<<c1.getr()<<endl;
    circle c2(c1);
    cout<<"c2:"<<c2.getr()<<endl;
    cout<<"c3:"<<getradius(c1)<<endl;
    circle c4(1);
    c4=fun1();
    cout<<"c4:"<<c4.getr()<<endl;
}
```

解答：

本题主要考查在什么情况下会调用拷贝构造函数。构造函数只在对象被创建时自动调用，而拷贝构造函数在下列三种情况下会被自动调用：

①用一个对象去初始化本类的另一个对象时。

②函数的形参是类的对象，在进行形参和实参的结合时。

③函数的返回值是类的对象，函数执行完返回时。

本题答案为：

```
c1:1
第 1 次调用拷贝构造函数!
c2:1
第 2 次调用拷贝构造函数!
c3:2
第 3 次调用拷贝构造函数!
c4:15
```

例题 8．读程序写结果或者程序填空。

```
#include <iostream>
using namespace std;
class A
{
    const int i;
    int &j;
public:
    A(int& var):i(10),j(var)
    {}
    void show()
    {
        cout<<"i:"<<i<<endl
            <<"j:"<<j<<endl;
    }
};
void main()
{
    int x=1;
    A a1(x);
    a1.show();
}
```

解答：

本题主要考查对符号常量和引用的理解。常量是不能被赋值的，一旦初始化后，其值就永不改变，引用变量也是不可重新指派的，初始化后，其值就固定不变了。
结果为：
i:10
j:1

例题 9．运行下列程序结果为_____。
```
#include <iostream>
using namespace std;
class Obj{
    static int i;
public:
    Obj(){i++;}
    ~Obj(){i--;}
    static int getVal(){return i;}
};
int Obj::i=0;
void f (){Obj ob2;cout<<ob2.getVal();}
int main(){
    Obj ob1;
    f();
    Obj*ob3=new Obj;cout<<ob3->getVal();
    delete ob3;cout<<Obj::getVal();
    return 0;
}
```
解答：
本题主要考查对静态数据成员的理解。在主函数中创建对象 ob1 则调用该类的构造函数，使得静态数据成员加 1，为 1；接着调用函数 f()，在函数中创建对象 ob2，这时再次调用构造函数，使得静态成员的值为 2，ob2.getVal()返回静态数据成员 i 的值，即输出 2。函数 f（）结束，则 ob2 的生存期结束，自动调用其析构函数使静态数据成员 i 的值变为 1。接着在主函数中用 new 运算符动态分配存储空间，又一次调用构造函数使 i 加 1，所以再次输出时 i 的值为 2。最后用 delete 释放 ob3 所指的对象空间，则会调用析构函数使 i 的值减 1，因此输出 i 的值为 1。本题答案为：221

例题 10．若类 A 是类 B 的友元，类 B 是类 C 的友元，则下列说法正确的是（　）。
(a)类 B 是类 C 的友元　　(b)类 A 是类 C 的友元
(c)类 A，B，C 互为友元　　(d)以上说法都不对
解答：
本题考查对友元关系的理解。友元关系是单向的，也是不能传递的。答案为：a

例题 11．当输入为 2　3 时，下列程序输出“两个数的和为：5”。请将程序补充完整。
```
#include <iostream>
using namespace std;
class num
{
    int x,y;
public:
    num(int=0,int=0);
    _____①_____
};
num::num(int x,int y)
{
    _____②_____

    _____③_____
```

```
}
int sum(num& n)
{
    return n.x+n.y;
}
void main()
{
    int i,j;
    cout<<"请输入两个数："<<endl;
    cin>>i>>j;
    ________④__________//定义对象 num1
    cout<<"两数的和为："<<sum(num1)<<endl;
}
```

解答：

本题主要考查友元的应用以及对不同作用域变量的引用方法。

答案为：①friend int sum(num&);②num::x=x;③num::y=y;④num num1(i,j);

【习题】

一、选择题

1．下列各项中不能用于声明类的成员访问控制权限的关键字是（　）。

(a)private　(b)protected　(c)public　(d)static

2．下列关于构造函数的说法错误的是（　）。

(a)构造函数的名字必须与类的名字相同。

(b)构造函数可以定义为 void 类型。

(c)构造函数可以重载、可以带有默认参数。

(d)构造函数可以由用户自定义也可以由系统自动生成。

3．有如下类声明：

```
class student
{
    int age;
    char *name;
};
```

则 student 类的成员 age 是（　）。

(a)公有数据成员　(b)私有数据成员　(c)保护数据成员　(d)私有成员函数

4．有如下类定义

```
#include <iostream>
using namespace std;
class point
{
int x,y;
public:
point():x(0),y(0){}
point(int x1,int y1=0):x(x1),y(y1){}
};
```

若执行语句

```
point a(2),b[3],*c;
```

则 point 类的构造函数被调用的次数是（　）。

(a)2 次　(b)3 次　(c)4 次　(d)5 次

5．在下列哪种情况下不会调用拷贝构造函数（　）。

(a)用一个对象去初始化本类的另一个对象时。

(b)函数的形参是类的对象，在进行形参和实参的结合时。

(c)函数的返回值是类的对象，函数执行完返回时。
(d)将类的一个对象赋值给另一个本类的对象时。
6．下列关于友元的描述错误的是（　）。
(a) 友元关系是单向的且不可传递
(b) 在友元函数中可以通过 this 指针直接引用对象的私有成员。
(c) 友元可以是一个普通函数也可以是一个类。
(d) 通过友元可以实现在类的外部对类的私有成员的访问。
7．有如下程序

```
#include <iostream>
using namespace std;
class AA{
    int n;
public:
    AA(int k):n(k){ }
    int get( ){ return n;}
    int get( )const{ return n+1;}
};
int main( )
{
    AA a(5);
    const AA b(6);
    cout<<a.get( )<<b.get( );
    return 0;
}
```

运行该程序结果为（　）。
(a)56　　(b)57　　(c)67　　(d)66
8．有如下程序：

```
#include   <iostream>
class Test {
public:
Test( ) { n+=2; }
~Test( ) { n-=3; }
static int getNum( ) { return n; }
private:
static int n;
};
int Test::n = 1;
int main( )
{
Test* p = new Test;
delete p;
cout << "n=" << Test::getNum( ) << endl;
return 0;
}
```

执行后的输出结果是（ ）。
(a) n=0　　(b)n=1　　(c)n=2　　(d)n=3
9．下列程序的运行结果为（　）。

```
#include <iostream>
class A
{
public:
    static int num;
    A& fun()
    {
        num++;
        return *this;
```

```
    }
};
int A::num=0;
void g(A& a)
{
    cout<<a.fun ().num<<endl;
}
void main()
{
    A a1;
    g(a1);
}
```

(a)0　　(b)1　　(c)2　　(d)3

10．运行下列程序结果为_____。

```
#include<iostream>
#include<iomanip>
using namespace std;
class MyClass{
public:
    MyClass(){cout<<'A';}
    MyClass(char c){cout<<c;}
    ~MyClass(){cout<<'B';}
};
int main(){
    MyClass p1,*p2;
    p2=new MyClass('X');
    delete p2;
    return 0;
}
```

执行这个程序幕上将显示输出_____。

(a)ABX　　(b)ABXB　　(c)AXB　　(d)AXBB

二、填空题

1．类的成员包括_____成员和成员_____。

2．释放对象所占的内存空间并完成善后处理工作的是_____函数。

3．拷贝构造函数以_____作为参数。

4．用指向对象的指针引用对象成员使用操作符_____。

5．当一个对象生成以后，系统就为这个对象定义了一个_____，它指向这个对象的地址。

6．在类中声明静态成员的关键字是_____。

7．非成员函数应声明为类的_____函数才能访问这个类的 private 成员。

8．C++建立和初始化对象的过程由_____完成。

9．对于常量数据成员和引用数据成员的初始化只能通过_____来完成。

10．在类中说明的具有类类型的成员称为_____。

11．下列为类的定义语句，是否有错，若有错请改正。

```
class circle ----------①
{
    double r=3;----------②
public:---------- ----------③
    circle(double i) ----------④
    {
        r=i; ----------⑤
    }
    double area();//面积----------⑥
```

```
        double prm();//周长-----------------------------⑦
        void printarea(double); -------------------------⑧
        void printprm(double); -------------------------
}-----------------------------------------------------⑩
//成员函数的实现
```

12．下列程序输出结果为 0,1.请将程序补充完整。

```
#include <iostream>
class A{
        int num;
public:
        A():num(0){}
        void set(int num)
        { _________①_____________}//给 A 的数据成员 num 赋值
        int get()
        {__________②_____________}//获得数据成员 num 的值
};
int main()
{
        A a;
        cout<<a.get()<<",";
        a.set(1);
        cout<<a.get()<<endl;
        return 0;
}
```

13．下列程序输出：2，33.4，tom。请将程序补充完整。

```
#include <iostream>
#include<string.h>
class A
{
        int i;
        float j;
        char c[20];
public:
        A(int x, float y, char ch[]):i(x),j(y)
        {
                _________①____________      //初始化成员 c 赋值
        }
        void printA()
        {
                _________②____________//输出三个私有数据成员
        }
};
void main()
{
        A a(2,33.4,"tom");
        a.printA();
}
```

14．下列程序的输出结果为

Object id=0
Object id=1
请将程序补充完整。

```
#include    <iostream>
class Point
{
public:
```

```
    Point(int xx=0, int yy=0) {X=xx; Y=yy; countP++; }
    Point( ) { countP--; }
    int GetX( ) {return X;}
    int GetY( ) {return Y;}
    static void GetC( ) {cout<<" Object id="<<countP<<endl;}
private:
    int X,Y;
    static int countP;
};
_____//静态数据成员的初始化
int main( )
{
    Point::GetC( );
    Point A(4,5);
    A.GetC( );
    return 0;
}
```

15. 插入排序算法的主要思想是：每次从未排序序列中取出一个数据，插入到已排序序列中的正确位置，InsertSort 类的成员函数 sort()实现了插入排序算法，请将画线处缺失的部分补充完整。

```
#include <iostream>
class InsertSort{
public:
    InsertSort(int*a0,int n0):a(a0),n(n0){}//a 是数组首地址，n 是数组元素个数
    void sort()
    {//此函数假设已排序序列初始化状态只包含 a[0]，未排序序列初始为 a[1]…a[n-1]
        for (int i=1;i<n;++i)
        {
            int j,t;
            for(_____;j>0;--j)
            {
                if(t>a[j-1])break;
                a[j]=a[j-1];
            }
            a[j]=t;
        }
    }
protected:
    int*a,n;//指针 a 用于存放数组首地址，n 用于存放数组元素个数
};
```

三、编程题

1. 自定义一个正方体类，它具有私有成员 x，表示正方体的每个面的正方形的边长。提供构造函数以及计算正方体的体积和表面积的公有成员函数，并编制主函数，对正方体类进行使用：说明正方体类对象，输入棱长，计算其体积和表面积并显示结果。

2. 设计一个时间类 Time，包括 3 个数据成员，时（hour）、分（minute）、秒（second），以及成员函数用于设置和读取时、分、秒，并按上午、下午各 12 小时或按 24 小时输出时间。

【参考答案】

一、选择题

1-5．dbbcd　　6-10．bbabd

二、填空题

1．数据、函数

2．析构

3．本类对象的引用

4．->

5．this 指针

6．static

7．友元

8．类的构造函数

9．成员初始化列表

10．对象成员

11．②⑩ 改正如下：

②double r;

⑩末尾加分号。

12．①A::num=num; ②return num;

13．

①strcpy(c,ch);

②cout<<i<<","<<j<<","<<c<<endl;

14．int Point::countP=0;

15． j=i,t=a[i]

三、编程题

1.参考程序如下：

```
#include <iostream>
#include<math.h>
class cube
{
	double x;
public:
	cube(double xx)
	{
		x=xx;
	}
	double volume();
	double sarea();

};
double cube::volume()
{
	return pow(x,3);
}
double cube::sarea()
{
	return pow(x,2)*6;
}
void main()
{
	double a;
	cout<<"请输入棱长：";
	cin>>a;
	cube c(a);
	cout<<"该正方体的体积为："<<c.volume ()<<endl
		<<"该正方体的表面积为："<<c.sarea()<<endl;
}
```

2.参考程序如下：

```
#include <iostream>
class Time
{
	int hour;
	int minute;
	int second;
public:
	int sethour(int);
	int setminute(int);
	int setsecond(int);
	int gethour();
	int getminute();
	int getsecond();
	void show12();
	void show24();
};
int Time::sethour(int h)
{
	if(h>=24||h<0)
		return 0;
	else
	{
		hour=h;
		return 1;
	}
}
int Time::setminute(int m)
{
	if(m>=60||m<0)
		return 0;
	else
	{
		minute=m;
		return 1;
	}
}
int Time::setsecond(int s)
{
	if(s>=60||s<0)
		return 0;
	else
	{
		second=s;
		return 1;
	}
}
int Time::gethour(){return hour;}
int Time::getminute(){return minute;}
int Time::getsecond(){return second;}

void Time::show12()
{
	if(hour==12)
		cout<<hour<<":"<<minute<<":"<<second<<"AM"<<endl;
	else
		cout<<hour%12<<":"<<minute<<":"<<second
```

```
                <<((hour<12&&hour>=0)?"AM":"PM")<<endl;
}
void Time::show24()
{
    cout<<hour<<":"<<minute<<":"<<second<<endl;
}

void main()
{
    int h,m,s;
    cin>>h>>m>>s;
    Time t;
    if(t.sethour(h)==0||t.setminute(m)==0||t.setsecond(s)==0)
    {
        cout<<"输入时间错误！";
        return;
    }
    t.show12();
    t.show24();

}
```