

中南大学c++题库

下列哪个是C++语言的合法的字符常量_____。 A " \0"

B ' 054' C ' \x89' D ' \092' 参考答案 C

定义整型变量x, y, z并赋初始值6的正确语句是_____。 A int

x=y=z=6; B int x=6, y=6, z=6; C x=y=z=6; D int x, y, z=6; 参考答案 B

在以下选项中, 与k=n++完全等价的表达式是_____。

A k=n, ++n; B n=n+1, k=n; C k=++n; D k+=n+1; 参考答案 A

表达式(1, 2, 3, 4)的结果是_____。

A 1 B 2 C 3 D 4 参考答案 D

下列字符串中可以用作C++标识符的是

A 2009var B goto C test-2009 D _123 参考答案 D

下列字符串常量中, 错误的表示是_____。

A " \ " Yes\ " or\ " No\ " " B "abcd\n" C ' ABC'

D "\tOK!" 参考答案 C

定义变量int a=6, 则a++的值是_____。

A 7 B 6 C 5 D 4 参考答案 B

已知字母A的ASCII码为65, 则表示字符常量A错误的是_____。

A '\101' B '\x41' C 65 D "A" 参考答案 D

下列常量中不能作为c++常量的是_____。

A 0382 B 0xB4 C .5e-2 D 123 参考答案 A

下列哪个是C++语言的不正确的标识符_____?

A _No1 B Ab1 C bgc D int 参考答案 D

关于C++和c语言的描述中, 错误的是_____。

A C语言是C++的一个子集 B C语言和C++是兼容的 C C++对C语言进行了一些改进 D C语言和C++都是面向对象的 参考答案 D

结构化程序由三种基本结构组成，不包括_____。

A 顺序结构 B 选择结构 C 控制结构 D 循环结构 参考答案 C

在位运算中，操作数每右移一位，其结果相当于_____。 A 操作数乘以2 B 操作数除以2 C 操作数乘以4 D 操作数除以4 参考答案 B

用c++语言编写的代码程序_____。

A 可立即执行 B 是一个源程序 C 经过编译即可执行 D 经过解释才能执行 参考答案 B

在函数中如果不说明变量的存储类型，则默认的存储类型是_____。

A auto B static C extern D register 参考答案 A

已知有定义

```
const int D 5;
```

```
int i=1;
```

```
double f=0.32;
```

```
char c=15;
```

则下列选项错误的是_____。

A ++i; B D--; C c++; D --f; 参考答案 B

在C++语言中，表示一条语句结束的标号是_____。

A “#” B “;” C “}” D “//” 参考答案 B

以下选项中属于c++语言的数据类型是_____。

A 复数型 B 日期型 C 双精度型 D 集合型 参考答案 C

二进制文件与字符文件不同的是_____。

A 二进制文件中每字节数据都没有用ASCII码表示 B 二进制文件包含了ASCII码控制符 C 二进制文件一般以字符‘\0’结束 D 二进制文件用字符endl表示行的结束 参考答案 A

变量生存期是指_____。

A 变量定义 B 程序的运行期间 C 变量在内存中分配存储单元到被释放期间 D 可执行代码的生存期 参考答案 C

x为int型，s为float型，x=3，s=2.5。表达式s+x/2的值为_____。

A 4 B 3.5 C 2.5 D 3 参考答案 B

不属c++源程序的翻译过程的是_____。

A 解释分析 B 词法分析 C 语法分析 D 链接 参考答案 A

x、y为整数，x=15，y=-2。表达式x>10 && y<2 || x*y==10 && x的值为()。

A 0 B 15 C 1 D 2 参考答案 C

下列关于c++源程序正确的说法是

A 每行只能写一个语句 B 每条语句(包括预处理命令)必须以分号结束 C 注释语句会生成机器码 D 最好向右缩进表达程序的层次结构 参考答案 D

c++源程序中，以下说法不正确的是_____。

A 逗号运算符的优先级最低 B Pcn和pcn是两个不同的标识符 C 变量定义语句一定会给变量赋初值 D 表达式1/4+1/4+1/4的结果为0 参考答案 C

不属于c++源程序的独立部分是_____。

A 声明部分 B 存储区分配部分 C 编辑预处理部分 D 函数部分 参考答案 B

函数的返回值的类型由_____决定。

A 调用该函数的主调函数类型 B return语句中的表达式类型 C 定义函数时所指定的返回值类型 D 调用函数时临时决定 参考答案 C

设有 `int a[10], *p=a` ; 则下面哪种表示与`a[5]`不等价: _____。

A `*(a+5)` B `*(p+5)` C `p[5]` D `p+5` 参考答案 D

若有: `int a,*p=&a`; 则下面哪个表达式为假_____。

A `*p= &a` B `p= &*p` C `a= *&a` D `p= &a` 参考答案 A

下面哪一C++ 语言表达式不能正确表示数学关系 $a < x \leq b$ 。选项为_____。

A `a<x&& x<=b` B `x<=b&& a<x` C `!(a>=x)&&!(x>b)` D `a<x<=b` 参考答案 D

关于源程序中注释部分的说法, _____是正确的。

A 注释参加编译, 出现在目标程序中 B 注释参加编译, 但不出现在目标程序中 C 注释不参加编译, 但出现在目标程序中 D 注释不参加编译, 也不出现在目标程序中 参考答案 D

下面字符序列中, C++合法的标识符是 _____。 A `abc.cpp`

B `#a` C `int` D `_12` 参考答案 D

下面字符序列中, 合法的常量是_____。

A `078` B `5L` C `9e7.5` D `'a1'` 参考答案 B

编译程序的功能是_____。

A 将汇编语言程序转换为目标程序 B 将汇编语言程序转换为高级语言程序 C 将高级语言程序转换为目标程序 D 将高级语言程序转换为汇编语言程序 参考答案 C

设`a`和`b`为整型变量, 执行语句`b=(a=2+3, a*4), a+5;`后`a`和`b`的值为_____。

A 5, 10 B 20, 25 C 5, 25 D 5, 20 参考答案 A

如果`a=1, b=2, c=3, d=4`, 则条件表达式“`a>b?a:c<d?c:d`” 的值为_____。

A 3 B 2 C 1 D 4 参考答案 A

C++源程序文件的扩展名为_____。 A .CPP B .C C .DLL
D .EXE 参考答案 A

n目运算符的含义是_____。

A 对n个数据进行操作 B 对n个数据进行加法 C 对n个类型进行整合
D 对3个类型进行转换 参考答案 A

以下属于c++语言的基本数据类型是_____。

A 复数型 B 集合型 C 整数型 D 频率型 参考答案 C

int a生存的自动变量a在没有赋值的情况下，其值为_____。 A 0
B -1 C 1 D 随机数 参考答案 D

有关符号常量不正确的说法是_____。

A 增加了程序执行速度 B 增强程序的可读性 C 书写简单，不易出错
D 修改程序方便 参考答案 A

_____不是Vc++6.0的开发过程。

A 创建工程 B 创建c++源程序文件 C 使用exe程序 D 编译源程序 参考
答案 C

vc++中常数0x11表示十进制数_____。

A 11 B 17 C 16 D 12 参考答案 B

vc++中常数020表示十进制数_____。

A 20 B 32 C 16 D 24 参考答案 C

下列标识符不正确的是_____。

A ints B _wqse12 C _12aas D ~awq 参考答案 D

下列标识符正确的是_____。

A 1asd B _swe C 'swe D @aws 参考答案 B

有关变量的正确说法是_____。

A 存储单元及内容 B 自动变化的数 C 地址 D 人为设置的数 参考

答案 A

int变量的字节长度为_____。

A 8 B 2 C 4 D 10 参考答案 C

C++是 _____ 。 A 面向任务的编程语言 B 面向过程的编程语言
C 面向过程和对象的混合编程语言 D 面向对象的编程语言 参考
答案 C

在C++中，运算对象必须为整型数的运算符是（ ）。

A % B / C * D %和* 参考答案 A

在以下选项中，可以用做变量名的是（ ）。

A 2 B sum C int D *p 参考答案 B

在下列选项中，非法的转义字符是（ ）。

A ''\037'' B ''\'' C ''\0x41'' D ''\t'' 参考答案 B

以下选项中，非法的表达式是（ ）。

A n=(i=2,++i) B j++ C ++(i+1) D x=j>0 参考答案 C

在C++中，属于合法字符常量的是（ ）。

A ''\084'' B ''\x43'' C ''ad'' D "\0" 参考答案 A

下列表达式中，可作为C++合法表达式的是：

A [4, 3, 2, 1] B (4, 3, 2, 1) C 4=3=2=1 D 4-- 参考答案 B

若有定义：int a=10;float x=2.6, y=5.1;则表达式x+a%3*(int)
(x+y)%3/5的值是：

A 2.8 B 2.6 C 2.4 D 3.0 参考答案 B

整型变量x和y的值相等，且不等于0，则以下表达式中结果为0的是（ ）。

A x || y B x | y C x & y D x ^ y 参考答案 D

下列选项中不属于C++的类型的是（ ）。

A. signed short int B unsigned long int C unsigned int D long short 参考答案 D

在C++语言中，080是（ ）。

A 十进制数 B 八进制数 C 十六进制 D 非法数 参考答案 D

在C++中，合法的长整型常数是（ ）。

A. 0L B 4962710 C 0.054838743 D 2.1869e10 参考答案 A

字符串常量“ME ”的字符个数是（ ）

A1 B 2 C 3 D 4 参考答案A

有如下表达式19/3*sqrt(4.0)/5，则该表达式值数据类型为（ ）。A. int B float C double D 不确定 参考答案 C

在C++中，合法的字符常量是（ ）。

A. '\084' B '\x43' C 'ah' D “0” 参考答案 B

以下程序段输出的结果是（ ）。char c=" \41" printf("%d", c);

A .A B 41 C 语法错 D 65 参考答案C

若d为double型变量，则表达式d=1, d+5, d++的值是（ ）。

A. 1 B 6.0 C 2.0 D 1.0 参考答案 D

整型变量a定义后赋初值的结果是（ ）。int a=2.8*6

A 12 B 16 C 17 D 18 参考答案 B

下列运算符中可用于实数的是（ ）。

A % B -- C << D & 参考答案 A

下列关于圆括号运算符的功能说法不正确的是（ ）。

A. 可用于强制类型转换 B 可用于类型构造 C 可用于类型声明

D 可用于函数调用 参考答案 C

设变量k为int型，下列选项中不正确的赋值表达式是（ ）。 A ++k
B n1 = n2 = n3 C k = 1 == 1 D a=b+c=1 参考答案 D

设int b=12表达式b+=b-=b*b求值后b的值是（ ）。

A. 552 B 264 C 144 D -264 参考答案 D

设int a=4, b=8; 则++a*b的结果是（ ）。

A 33 B 40 C 32 D 41 参考答案 B

关于C++和C语言的描述中，错误的是（ ）。

A C语言是C++的一个子集 B C语言和C++是兼容的 C C++对C语言进行了一些改进 D C语言和C++都是面向对象的 参考答案 D

若w、x、y、z、m均为int型变量，则执行下面的语句后m的值是（ ）。

```
w=2, x=3, y=4, z=5;
```

```
m=(w<x)?w:x;
```

```
m=(m<z)?m:z;;
```

```
m=(m<y)?m:y;
```

A 2 B 3 C 5 D 4 参考答案 A

下列描述正确的是（ ）

A 不是每个C++程序都必须包含预处理命令#include<iostream.h>。
B C++程序必须包含return语句。 C C++程序中的变量必须定义在主程序的内部。 D C++程序中的符号变量可以不进行定义。 参考答案 A

以下选项中，合法的标识符是（ ）

A long B _2Test C 3Dmax D A.dat 参考答案 B

在C++语言中，自定义的标识符号

A 能使用关键字并且不区分大小写。 B 不能使用关键字并且不区分大小写。 C 能使用关键字并且区分大小写。 D 不能使用关键字并且不区分大小写。 参考答案 D

下列符号中，不是分隔符的是

A ; B // C , D {} 参考答案 B

下列运算符中优先级最低的是

A & B && C ^ D | 参考答案 B

将小写字母n赋给字符变量onechar，正确的操作是 ()

A onechar=' \n' B onechar=" n" C onechar=110

D onechar=' N' 参考答案 C

能正确表示逻辑关系“a大于等于10或a不大于0”的C++语言表达式是 ()

A a>=10 or a=0 B a>=0 or a<=10 C a>=10 && a<=0 D a>=10 || a<=0 参考答案 D

已知大写字母A的ASCII码值是65，小写字母a的ASCII码是97，则用八进制表示的字符常量“\101”是

A 字符A B 字符a C 字符e D 非法的常量 参考答案 B

以下非法的赋值语句是

A n=(i=2, ++i); B j++; C ++(i+1); D x=j>0; 参考答案 C

下面正确的赋值语句是

A a=7+b+c=a+7; B a=7+b++=a+7; C a=7+b, b++, a+7; D a=7+b, c=a+7; 参考答案 A

语句cout<<(a=2)&&(h=-2); 的输出结果是

A 无输出 B 编译错误 C -1 D 1 参考答案 D

C++语言中在C语言的基础上增加的特性不包括

A 结构 B 引用 C 类和对象 D 静态成员函数 参考答案 A

int x=3, y=4, z=5; 则值为0的表达式是 ()

A x>y++ B x<=++y C x!=y+z>y-z D y%z>=y-z 参考答案 A

对字符常量与字符串常量的描述不正确的是（）：

A 两者表示形式不同：前者使用单引号，后者使用双引号。 B 存放不同：前者存放在字符变量中，后者存放在字符数组或字符指针指定的位置。 C 存放字符串常量时系统会自动加一个结束符''\0'' D 字符、字符串都能参与连接运算。 参考答案 D

Visual C++6.0中打开一个已存在的项目，应打开其中后缀名为（）的文件

A htm B h C dsw D cpp 参考答案 C

所有的表达式都有值。 [参考答案] 正确

包含单个字符的字符串常量和字符常量是相同的。 [参考答案] 错误

空字符串和空格串是一样的。 [参考答案] 错误

若x为整型变量，j为实型变量，当执行x=(int)j;语句后，j也变为整型变量。 [参考答案] 错误

自减运算符“--”可以用于变量，也可以用于常量。

[参考答案] 错误

字符型数据可以和整型数据之间通用。 [参考答案] 正确

若表达式(x+(y-z)*(m/n))+3中的变量均为double型，则表达式值的类型为__1__。

参考答案 1: double类型或double型或双精度或双精度类型或双精度型或double

有如下循环语句：

```
for(int i=50; i>20; i-=2) cout<<i<<',';
```

运行时循环体的执行次数是__1__次。

参考答案 1: 15

```
#include <iostream.h> void main() { char k='c',h='d'; int i=1,j=2; k+= i++; h-=--j; cout<<k<<h<<i<<j<<endl; } 程序的
```

运行结果：__1__。 参考答案 1: dc21

```
#include <iostream.h> void main() { int x,y,n; x=y=1; n=-x&&++y; cout<<n<<" "<<x<<" "<<y<<endl; n=(--x) || (++y); cout<<n<<" "<<x<<" "<<y<<endl; }
```

程序运行结果 第一行为：__1__。 第二行为：__2__。

参考答案 1: 0 0 1 2: 1 -1 1

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

```
void main( ) { int m=18,n=3; float a=27.6,b=5.8,x;  
x=m/2+n*a/b+1/4; cout<<x<<endl; }
```

程序运行结果是：__1__。

参考答案 1: 23.2759

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

```
void main( )
```

```
{ int a=6,b=13;
```

```
cout<<(a+1,b+a,b+10)<<endl;
```

```
}
```

写出它们的输出结果 __1__。 程序运行后，变量a的值__2__。

参考答案 1: 23 2: 6

有如下程序段：

```
int x=1, y=2, z=3;
```

```
X=x^z;
```

```
y=y^z;
```

```
z=x^y;
```

```
cout<<x<<y<<z;
```

执行这个程序段的输出是__1__。

参考答案 1: 2

有如下程序段:

```
for (int i=1; i<=50; i++) {  
    if(i%3 !=0)  
        continue;  
    else  
        if(i%5 !=0)  
            continue;  
    cout << i << ", ";  
}
```

执行这个程序段的输出是__1__。

参考答案 1: 15, 30, 45

设int x, y;

则执行y=(x=1, ++x, x+2);语句后, y的值是__1__。

参考答案 1: 4

x、y、z为float, 下列公式的赋值表达式为__1__。

参考答案 1: $z = (x*y >= 0) ? (x+y) * (x-x*y) / (x*x+y*y) / x : (x+y) * (y-x*y) / (x*x-y*y) / y$

标识符是以__1__及下划线开头的数字、字母及下划线组成字符串。

参考答案 1: 字母

通过文字编辑建立的c++源程序的扩展名为__1__。

参考答案 1: .cpp 或 cpp

给变量赋初值，初值必须是_____1__和常数表达式。

参考答案 1：常数

自加和自减的两种形式是__1__和后缀。

参考答案 1：前缀

修饰int的关键字有__1__、short和long。

参考答案 1：unsigned

c语言是__1__的程序设计语言，c++是面向过程和对象的程序设计语言。

参考答案 1：面向过程

n个字符的字符串存储占用的字节数为__1__，结尾的ascii码值为0。

参考答案 1：n+1

关系运算符、逻辑运算符和算术运算符的计算次序为：__1__

参考答案 1：先算术运算符再关系运算符最后逻辑运算符

写出三种逻辑运算符：__1__、__2__和__3__。

参考答案 1：|| 2： && 3：!

写出三个2目位运算符：__1__、__2__和__3__。

参考答案 1：& 2：^ 3：|

可存储小数的基本数据类型为__1__和双精度实型。

参考答案 1：单精度实型 或 float

变量的基本类型有__1__、实型、字符型和枚举型。

参考答案 1：整型

vc++源程序文件的扩展名为__1__，可执行程序的扩展名为exe。

参考答案 1：.cpp

变量的3要素是类型、__1__和变量值。

参考答案 1: 名 或 变量名

指数形式的浮点常数格式为: __1__。

参考答案 1: $Ae \pm B$

变量的存储类型有自动存储类、__1__、静态存储类和外部存储类。参
考答案 1: 寄存器存储类

Visual C++6.0的集成开发环境(IDE)集编辑、__1__、链接和调试运行
于一体。

参考答案 1: 编译

应用程序hello.cpp中只有一个函数, 这个函数的名称是__1__。(注
意: 答案只写函数名)

参考答案 1: main

若a为整型变量, 且a的初值为6, 则计算表达式 $a+=a-=a*a$ 后, a的值
为__1__。

参考答案 1: -60

表达式 ~ 10 的值为__1__。

参考答案 1: -11

若x和y均是整型变量, 当执行 $y=(x=1, ++x, x+2);$ 语句后, y的值是
__1__。

参考答案 1: 4

若有定义语句: $\text{int } a=1, b=2, c=3;$ 则表达式 $c+=a>b?++a:++b$ 的值为
__1__。

参考答案 1: 6

表达式 $!-5+31>>2$ 的值为__1__。

参考答案 1: 7

C++中的一个三目运算符是__1__。

参考答案 1: ?:

C++中的字符串的结束符是__1__。

参考答案 1: ‘\0’

C++中的标识符只能由字母、数字和__1__组成。

参考答案 1: 下划线

十六进制常量前面应该加__1__。

参考答案 1: 0x;0X

在C++结构化程序设计框架中，__1__是程序的基本组成单元。

参考答案 1: 函数

假设int a=1, b=2;则表达式a+++--b的值为__1__.

参考答案 1: 2

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

```
void main()
```

```
{
```

```
    int num=0; i=8;
```

```
do{
```

```
    i--;
```

```
    num++; }
```

```
while(--i):
```

```
cout<<num<<endl;
```

```
}
```

参考答案 4

有如下程序：

```
# include <iostream>
# include <iomanip>
void main( )
{
    int s[ ]={123, 234};
    cout<<right<<setfill('*')<<setw(6);
    for(int i=0; i<2; i++)
    {
        cout<<s[i]<<endl;
    }
}
```

运行时的输出结果是()。

A 123 234 B ***123 234 C ***123 ***234 D ***123 234*** 参考答
案 B

有如下程序：

```
# include <iostream>
using namespace std;
class Publication{ //出版物类
char name[30];
public:
Publication(char *name="未知名称"){
strcpy(this->name, name);
```



```

}

const char * getName()const{ return name; }

virtual const char * getType()const{ return "未知类型";}

};

class Book: public Publication{ //书类
public:
    Book(char *name): Publication(name) {}
    virtual const char * getType()const{ return "书";}
};

void showPublication( Publication &p) {
    cout<<p.getType()<<" : "<<p.getName()<<endl;
}

int main() {
    Book book("精彩人生");
    showPublication(book);
    return 0;
}

```

运行时的输出结果是

A 未知类型：未知名称 B 未知类型：精彩人生 C 书：未知名称
D 书：精彩人生 **参考答案 D**

有如下程序：

```

# include <iostream>

using namespace std;

```

```

class A {
public:
    A(int i) : r1(i) {}
    void print() {cout<<'E'<<r1<<'-' ;}
    void print() const {cout<<'C'<<r1*r1<<'-' ;}
private:
    int r1;
};

int main() {
    A a1(2); const A a2(4);
    a1.print(); a2.print();
    return 0;
}

```

运行时的输出结果是

A 运行时出错 B E2-C16- C C4-C16- D E2-E4- **参考答案 B**

有如下程序段：

```

    int i=1;
    while(1)
    {    i++;
        if (i==10) break;
        if (i%2==0)    cout<<'*' ;
    }

```

执行这个程序段输出字符*的个数是()。

A 10 B 3 C 4 D 5 **参考答案 C**

有如下程序： `int x=3; do { x-=2; cout<<x; } while(!(--x))`；执行这个程序的输出结果是（ ）。

A 1 B 3 0 C 1-2 D 死循环 参考答案 C

有如下语句序列： `int k=0; do { k+=5; cout<<' $' ; }while(k<19); while(k-->0) cout <<' *' ;` 执行上面的语句序列输出字符“\$”和“*”的个数分别是（ ）。

A 4和20 B 5和20 C 4和21 D 5和21 参考答案 A

已知语句：

`while(!x)`

其中表达式!x等价于_____。

A x=1 B x!=0 C x==0 D x!=1 参考答案 C

C++语言的跳转语句中，对于break和continue说法正确的是_____。

A break语句只应用于循环体中 B continue语句只应用于循环体中
C break是无条件跳转语句，continue不是 D break和continue的跳转范围不够明确，容易产生问题 参考答案 B

在下面的循环语句中循环体执行的次数为_____。

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

`if(i>n/2)break;`

A n/2 B n/2+1 C n/2-1 D n-1 参考答案 B

下面for语句的循环次数为_____。

`for(int i=0,x=0;!x&& i<=5;i++);` A 5 B 6 C 7 D 无穷次

参考答案 B

与下面程序段等价的是_____。

`while(a)`

`{ if(b)continue;`

c;

} A while(a) {if(!b)c;} B while(c) {if(!b)break;c;} C while(c) {if(b)c;} D while(a) {if(b)break;c;} 参考答案 A

已定义int a,b; 下列switch语句中格式正确的是 。

A. switch(a) { case b+1:a--;break; case b+2:a++;break; }

B. switch(a==b) { default:a-b}

C. switch(a/10+b) { case 5:a-b; default:a+b; }

D. switch(a*a) { case 1,2:++a;case 3,4:++b;} 参考答案 C

在下列程序段中，两个cout输出语句()。

```
while (2>3)
```

```
{ cout<<"I ' m a student. " <<endl; }
```

```
do
```

```
{ cout<<"I 'm a student. " <<endl; } while(2>3);
```

A 都有输出。 B 只有第1个有输出。 C 只有第2个没有输出。 D 第1个没有输出，第2个有输出。 参考答案 D

下列语句中错误的是()。

A if (a>b) cout<<a; B if (&&) ; a=m; C if (1) a=m;

else a=n; D if (a>0) ; else a=n; 参考答案 B

在C++语言中，所谓选择结构，就是按照()有选择地执行程序中的语句。

A 给定数值 B 给定符号 C 给定程序 D 给定条件 参考答案 D

while后面的“条件表达式”一定要用一对 括起来。

A 防括号 “[]” B 圆括号 “()” C 花括号 “{ }” D 引号 “ ” 参考答案 B

已知语句：

```
while(!E);
```

其中表达式!E等价于_____。 A E==0 B E!=0
C E!=1 D E=1 参考答案 A

在C++程序中，一个标识符在使用前必须先_____后引用。

A 声明 B 设定 C 定义 D 调用 参考答案 C

下列语句段将输出字符'*'的个数为_____。

```
int i=100;
```

```
while(1)
```

```
{
```

```
  i--;
```

```
  if(i==0)break;
```

```
  cout<<'*';
```

```
}
```

 A 98 B 99 C 100 D 101 参考答案 B

若i为整型变量，则以下循环的执行次数是_____。

```
for(i=2;i==0;) cout<<i<<endl;
```

A 0 B 1 C 2 D 无限次 参考答案 A

for(int i=0;x<10;x++) x++; 语句的循环次数是_____。 A 10
B 9 C 4 D 5 参考答案 D

循环体被至少执行一次的循环语句为_____。

A while B for C do ...while D 多重循环 参考答案 C

执行语句序列：

```
int i=0;
```

```
while(i<25)
```

```
i+=3;
```

```
cout<<i;
```

输出结果是 。

A 24 B 25 C 27 D 28 参考答案 C

执行语句序列：

```
int x=3;
```

```
do
```

```
{
```

```
x-=2;
```

```
cout<<x;
```

```
}while(!(--x));
```

输出结果是 。

A 1-2 B 1 C 30 D 死循环 参考答案 A

执行语句序列：

```
int n;
```

```
cin>>n;
```

```
switch(n)
```

```
{
```

```
    case 1:
```

```
    case 2: cout<<'1' ;
```

```
    case 3:
```

```
    case 4: cout<<'2' ; break;
```

```
    default: cout<<'3' ;
```

```
}
```

时，若从键盘上输入1，则屏幕显示()。 A 1 B 2 C 3

D 12 参考答案 D

假定所有变量均已正确说明，下列程序段运行后，x的值是()。

```
a=b=c=0;
```

```
x=35;
```

```
if (!a) x--;
```

```
else if (b);
```

```
        if (c) x=3;
```

```
        else x=4;
```

A 34 B 4 C 35 D 3 参考答案 B

下列程序段的输出是()。

```
int a=2, b=-1, c=2;
```

```
if(a<b)
```

```
    if(b<c)
```

```
        c=0;
```

```
    else
```

```
        c+=1;
```

```
cout<<c<<endl;
```

A 0 B C 2 D 3 参考答案 C

下列关于循环的描述中，错误的是()。

A do-while、while和for循环中的循环体均可以由空语句组成 B while循环是先判断表达式，后执行循环体语句 C do-while、while和for循环均是先执行循环体语句，后判断表达式 D do-while循环体至少无条

件执行一次，而while循环体可能一次也不执行 参考答案 C

下列关于for循环的描述中，正确的是()。

A for循环只能用于循环次数已经确定的情况 B for循环的循环体语句中，可以包含多条语句，但必须用花括号{ }括起来 C for循环是先执行循环体语句，后判断表达式 D 在for循环中可使用continue语句结束循环，接着执行for语句的后继语句 参考答案 B

下列关于break语句的描述中，不正确的是 。 A break语句可用在循环体中，它将使执行流程跳出本层循环体 B break语句在一个循环体内可以出现多次 C break语句可用在switch语句中，它将使执行流程跳出当前的switch语句 D break语句可用在if语句中，它将使执行流程跳出当前的if语句 参考答案 D

下列关于do-while语句的描述中，正确的是 。

A do-while语句所构成的循环只能用break语句跳出 B do-while语句所构成的循环不能用其他语句构成的循环来代替 C do-while语句所构成的循环只有在while后面的表达式非零时才结束 D do-while语句所构成的循环只有在while后面的表达式为零时才结束 参考答案 D

为了避免嵌套的条件语句的二义性，C++语言规定else与 ()配对。

A 编辑时在同一列的if B其之前最近的还没有配对过的if C 其之后最近的if D同一行上的if 参考答案 B

下列关于switch语句的描述中，正确的是 。 A switch语句中case子句后面的表达式只能是整型表达式 B switch语句中每个语句序列中必须有break语句 C switch语句中default子句可以没有，也可以有一个 D switch语句中default子句只能放在最后 参考答案 C

下列选项中属于C++语句的是()。 A ; B a=87 C i+5

Dcout<<' \n' 参考答案 A

若有定义int i=1; 则执行完循环while(i++<5);以后i的值为_____。

A 6 B 5 C 4 D 7 参考答案 A

若已有 int e; 则while(e) {…}, 中的循环控制条件等价于:

_____。

A $e = 0$ B $e \neq 0$ C $e = 1$ D $e \neq 1$ 参考答案 B

下列说法不正确的是：_____。

A for, while 和 do-while 循环体中的语句可以是空语句。 B 使用 while 和 do-while 循环时，循环变量初始化的操作应在循环语句之前完成。 C for 和 do-while 循环都是先执行循环体语句，后判断循环条件表达式。 D while 循环是先判断循环条件表达式，后执行循环体语句。

参考答案 C

关键字 break 不会直接影响的是_____。

A if B for C switch D while 参考答案 A

与 switch 同时出现的关键字是_____。

A define B typedef C do D case 参考答案 D

下面_____不是必须的 C++ 源程序格式要求。

A 语句以分号结束 B 编译命令不要以分号结束 C 函数体放在 {} 内
D {} 内的语句向右缩几个字符 参考答案 D

C++ 源程序分为_____区。

A 1 个 B 2 个 C 包含和定义及条件编译 3 个 D 4 个 参考答案 B

设 x、y、z、t 均为 int 变量，则执行以下语句后，t 的值为 ()。

```
x=y=z=1;
```

```
t=++x || ++y && ++z;
```

A

不定值

B 2 C 1 D 0 参考答案 C

执行 `if((x=y=2)>=x&&(x=5)) y*=x;` 语句后，变量 x、y 的值分别为 ()。

A 2、4 B 5、2 C 5、10 D 执行时报错 参考答案 C

设 x、y 和 z 是 int 型变量，且x=3、y=4、z=5，则下列表达式中，其值为 0 的是（ ）。

A x&&y B x<=y C x||y+z&&y-z D !((x<y)&&!z||1) 参考答案 D

逻辑运算符两侧的运算对象的数据类型是（ ）。

A 只能是 0 或 1 B 只能是 0 或非 0 正数 C 只能是整型或字符型数据 D 可以是任何类型的数据 参考答案 D

与y=(x>0?2:x<0?1:0);的功能相同的语句是（ ）。

A if (x>0) y=2; else if(x<0) y=1; else y=0;

B if (x) if(x>0) y=1; else if(x<0) y=2; else y=0;

C y=1; if (x) if (x>0) y=2; else if(x==0)y=0; else y=1;

D if(x<0) y=2; if(x>0) y=1; else y=0; 参考答案 A

已知 year 为整型变量，不能使以下表达式为“真”的数据是（ ）。

(year%4==0&&year%100!=0)||year%400==0

A 1990 B 1992 C 1996 D 2000 参考答案 A

当把下面4个表达式用作if语句的条件表达式时，有一个选项与其它3个选项含义不同，这个选项是（ ）

A k%2 B k%2==1 C (k%2)!=0 D !k%2==1 参考答案 D

为了避免在嵌套的条件语句if..else中产生二义性，C++语言规定：else语句总是与（ ）配对。

A 缩排位置相同的if B. 前面最近的if C 后面最近的if

D 前面最近未配对的if 参考答案 D

下面的for语句的循环次数为（ ）。

for(x=1, y=0; (y!=19&&(x<6); x++);

A 无限循环 B 循环次数不定 C 最多执行6次 D 最多执行5次
参考答案 D

C++语言中while循环和do...while循环的主要区别是（ ）。

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的循环控制条件严格 C do...while允许从外部转到循环体内 D do...while的循环体不能是复合语句 参考答案 A

有以下程序：#include<iostream>

```
using namespace std;
```

```
void main() {
```

```
int a=5, b=4, c=3, d=2;
```

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

```
cout<<d<<endl;
```

```
else if ((c-1>=d)==1)
```

```
cout<<d+1<<endl;
```

```
else
```

```
cout<<d+2<<endl;
```

```
}执行后输出结果是（ ）。
```

A 2 B 3 C 4 D 编译时有错，无结果 参考答案 B

t为int类型，进入下面的循环之前，t的值为0。对下面的语句

```
while(t=1){...}
```

则以下叙述中，正确的是（ ）。

A 循环控制表达式的值为0 B 循环控制表达式的值为1 C 循环控制表达式不合法 D 以上说法都不对 参考答案 B

若有条件语句“if(exp) a++ else b--”，则以下表达式中，能完全等价于表达式(exp)的是（ ）。

A (exp==0) B (exp!=0) C (exp==1) D (exp!=1) 参考答案 B

以下程序段的输出结果是（ ）。

```
int x=5;
do{
cout<<x--;}
while(!x);
```

A 54321 B 43210 C 5 D 4 参考答案 C

用于实现程序流程转移的语句是（ ）

A 选择语句 B .跳转语句 C 声明语句 D循环语句 参考答案 B

下列关于switch的描述正确的是（ ）

A switch语句中default子句可以没有，也可以有一个。 B switch语句中每个语句序列中必须有break语句。 C switch语句中default子句只能放在最后。 D |switch语句中case子句最后的表达式只能是整型表达式。 参考答案 A

C++语句中while循环和 do while循环的主要区别是（ ）

A do while的循环体至少无条件的执行一次。 B while 的循环控制条件比do while的循环控制条件严格。 C. do while 允许从外部转到循环体内。 D do while的循环体不允许是复合语句。 参考答案 A

下列声明语句没有起到定义作用的是（ ）

A int count; B const double pi=3.1415; C int max(int a,int b) {return a>b ?a:b;} D extern long index; 参考答案 D

下列程序段循环了（ ）次。

```
int x=-9;
while(++x) {}
```

A 8 B 9 C 10 D 无限 参考答案 A

下列不是循环语句的是

A while语句 B do...while语句 C for语句 D if...else语句 参考答案 D

对于语句for(x=0, y=0; (y!=123)&&(x<4); x++); 正确的说法是 A 是无限循环 B 循环次数不定 C 循环4次 D 循环3次 参考答案 C

有以下程序段

```
int k=0;
```

```
while(k=1)k++;
```

while循环执行的次数是

A

无限次(死循环)

B 有语法错, 不能执行 C 一次也不执行 D 执行1次 参考答案 A

设x, y, z, k都是int型变量, 则执行表达式: x=(y=4, z=16, k=32)后, x的值为 ()

A 4 B 16 C 32 D 52 参考答案 C

若k为int 类型, 且k的值为3, 执行语句k+=k-=k*k后, k的值为 ()

A -3 B 6 C -9 D -12 参考答案 D

使用cin>>输入多个数据时, 输入时各数据之间所用的分隔符不能是 ()

A 空格键 B Tab键 C 逗号 D 回车符 参考答案 C

任何一个表达式加上一个分号就构成一个语句。[参考答案] 正确

在逻辑表达式中, 有些逻辑运算符可能不会被执行到。

[参考答案] 正确

条件运算符的结合方向为从左到右。[参考答案] 错误

在switch语句中，switch后面的表达式可以为任意类型。

[参考答案] 错误

在C++中，空语句表示什么都不做，因此毫无意义。

[参考答案] 错误

while语句的循环体有可能一遍都不执行，而do--while语句至少执行一遍循环体 [参考答案] 正确

以下程序输入两个正整数，求它们的最大公约数。

```
#include <iostream.h>

void main( )
{
    int m,n,r;
    cin>>m>>n;
    r=m%n;
    while( __1__)
    {
        m=n;
        n=r;
        __2__ ;
    }
    cout<<"最大公约数为："<<n<<endl;
}
```

填空完成该程序。

参考答案 1: $(r=m\%n) != 0$ 或 $r != 0$ 或 r 2: $r=m\%n$ 或 $r=m-m/n*n$

以下程序显示输出倒三角图案。

*

```
#include <iostream.h> void main( ) { int i, j;
for(i=0; i<4; i++) { for(j=0; j<i; j++) cout<<"
"; for(j=0; j<2*(3-i)-1; j++) cout<<"__1__";
cout<<"__2__"; } } 填空以完成该程序。
```

参考答案 1: "*" 或 '*' 2: endl 或 '\n' 或 "\n"

以下程序计算表达式 $1/n(1^2+2^2+\dots+n^2)$ 的值。 #include <iostream.h> void main() { int n, k=1, sum=0, mean; cout<<"输入n的值: "; cin>>n; while(k<=n) { sum=sum+k*k; __1__; } if (n>0) mean=sum/n; else mean=0; cout<<"表达式的值="<<mean<<endl; } 填空以完成该程序。

参考答案 1: k++ 或 k=k+1

根据输入的年、月，判断该月的天数。例如，2008年2月的天数为29天，2010年5月为31天。完成以下程序： #include <iostream.h> void main() { int year, month, days; 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) || !(year%4)&& year%100) __1__ ; else days=28; } cout<<year<<"年"<<month<<"月的天数为: "<<days<<endl; }

参考答案 1: days=29

输入任意一个字符，如果该字符是英文字母，则输出其ASCII码。 void main() { char c; cout<<"请输入一个字符: "; cin>>c; if

```
((c>='A' && c<='Z') || (c>='a' && c<='z'))    cout<<(int)c<<endl; }
```

如输入D d M n, 则输出为: __1__ (写在一行, 以空格分隔)

参考答案 1: 68 100 77 110

符合结构化原则的三种基本控制结构是: 选择结构、循环结构和 __1__。

参考答案 1: 顺序结构

Break语句用于结束整个循环, __1__ 语句用于结束本次循环。

参考答案 1: continue

完成从键盘上输入100个数, 用选择法按降序排序输出的程序。

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

```
Main()
```

```
{ int i, j, a[100], tem;
```

```
  for (i=0; i<100; i++)
```

```
  { cout<< "输入第" << i+1<< "个数: ";
```

```
    Cin>>a[i];
```

```
  }
```

```
  for (i=0; i<99; i++)
```

```
    for(j=__1__
```

```
; j<=99; j++)
```

```
      if (__2__
```

```
  ])
```

```
    {tem=a[i]; a[i]=a[j]; a[j]=tem;}
```

```
  for (i=0; i<100; i++)
```



```
    cout<<a[i]<<"  ";  
}
```

参考答案 1: i+1 2: a[i]<a[j]

语法的三种基本结构为顺序结构、__1__和循环结构。

参考答案 1: 选择结构

一个数如果等于它的因子（包括1）之和，则称该数为完数，如：
6=1+2+3；编程求2到100之间 的完数。填写空缺部分。

```
#include <iostream>  
  
using namespace std;  
  
int main( )  
{int m, i, s=0;  
    for(m=2;m<100;m++)  
    {__1__ ;  
        i=1;  
        while( __2__ )  
            {if(m%i==0) s=s+i;  
                i++;  
            }  
        if( __3__ )  
            cout<<"  "<<s<<" \t";  
    }  
    cout<<endl;
```

```
return 0;  
}
```

参考答案 1: s=0 2: i<m或m>i 3: s==m或 m==s

设有: int y=10;

y+=y-=y-y; 则 y 的值是: __1__。

参考答案 1: 20

三种基本的程序控制结构是: 顺序结构, 选择结构, __1__。

参考答案 1: 循环结构

在switch语句中, 每个语句标号所含关键字case后面的表达式必须是 __1__。

参考答案 1: 常量表达式

输入一个正整数, 求该数的阶乘。

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i,n;
```

```
    long int fact=1;
```

```
    cout<<"请输入一个正整数:";
```

```
    __1__;
```

```
    for(i=2;i<=n;i++)    __2__*=i;
```

```
    cout<<n<<"!="<<fact<<endl;
```

```

        return 0;
    }
}

```

参考答案 1: `cin>>n 2: fact`

用“枚举法”求解百元买百鸡问题。假定公鸡5元1只，母鸡3元1只，小鸡1元3只，现在有100 元钱要买100只鸡，且需包含公鸡、母鸡和小鸡，编程列出所有可能的购鸡方案。

```

#include <iostream>

using namespace std;

int main()
{
    int x,y,z;
    cout<<"公鸡数\t"<<"母鸡数\t"<<"小鸡数\t"<<endl;
    for(x=1;x<=19;x++)
        for(y=1;y<=31;y++)
        {
            z=__1__;

            if(__2__==100)
                cout<<x<<' \t'<<y<<' \t'<<z<<endl;
        }
    return 0;
}

```

参考答案 1: `100-x-y` **2:** `5*x+3*y+z/3.0`

求“水仙花数”。所谓“水仙花数”是指一个三位正整数，其各位数字

的立方和等于该数本身。例如：153=1³ +5³ +3³ 。

```
#include <iostream>

using namespace std;

int main()
{
    int i, j, k, n;
    cout<<"水仙花数： ";
    for(n=100;n<1000;n++)
    {
        i=__1__
;        // i为百位数字
        j=n/10-i*10;           // j为十位数字
        k=__2__;                // k为个位数字
        if(i*i*i+j*j*j+k*k*k==n)
            cout<<n<<' ';
    }

    cout<<endl;

    return 0;
}
```

参考答案 1: n/100 2: n%10

输入x，计算sin(x)。计算公式为： $\sin(x) = x - \frac{x^3}{3!} + \frac{(-1)^{(n-1)}x^{(2n-1)}}{(2n-1)!} + \dots$

```
#include <iostream>
```

```

#include <cmath>
using namespace std;
int main()
{
    int n=1;
    double x,t,sinx(0);
    cout<<"请输入x的值: ";
    cin>>x;
    t=__1__;
    while(fabs(t)>=0.000001)
    {
        sinx+=t;
        t=-t*x*x/(__2__
);
        n+=2;
    }
    cout<<"编程求得的sin("<<x<<")="<<sinx<<endl;
    cout<<"调用标准函数求得的sin("<<x<<")="<<sin(x)<<endl;
    return 0;
}

```

参考答案 1: x 2: (n+1)*(n+2)

一个小球从100米高处落下，每次落地后反弹回原高度的一半，再落下。求它在第10次落地时 共经过多少米？第10次反弹多高？

```

#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    float s=100.0, h=s/2;
    for(int i=2; __1__ ; i++)
    {
        s+=h*2;
        h=__2__ ;
    }

    cout<<"第10次落地时，共经过"<<s<<"米；反弹高度是"
    <<h<<"米"<<endl;

    return 0;
}

```

参考答案 1*i*<=10或10>=*i*或*i*<11或11>=*i* 2: h/2或h/2;

显示输出3~100之间的所有素数。

```

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

```

```

cout<<"3~100之间的素数是："<<endl;
for(int m=3;m<100;m+=2)
{
    int k=int(sqrt(m));
    for(int i=2;i<=k;i++)
        if(m%i==0) __1__ ;

    if(__2__ )
        cout<<m<<" ";
}

cout<<endl;

return 0;
}

```

参考答案 1: break 2: k<i 或 i>k

输入一个自然数，将该数的每一位数字按反序输出。例如：输入12345，输出54321。

```

#include <iostream>

using namespace std;

int main()
{
    unsigned long int num,digital;

    cout<<"请输入一个自然数:";

    cin>>num;

```

```

do
{
    digital=__1__
;
    num/=10;
    cout<<__2__;
}while(num>0);
cout<<endl;
return 0;
}

```

参考答案 1: num%10 2: digital

求出满足不等式 $1+1/2+1/3+\dots+1/n \geq 5$ 的最小n值。

```

#include <iostream>
using namespace std;
int main()
{
    int i=0;double s=0;
    while(__1__)

        s+=double(1)/__2__;
    cout<<"n="<
return 0;

```



```
}
```

参考答案 1: s<5 2: ++i

求自然数1~100之和，即计算sum=1+2+3+...+100。

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i(1), sum(0);
```

```
    while(__1__)
```

```
    {
```

```
        __2__;
```

```
        i++;
```

```
    }
```

```
    cout<<"sum="<<sum<<endl;
```

```
    return 0;
```

```
}
```

参考答案 1: i<=100或100>=i 2sum+=i 或sum=sum+i 或sum=i+sum

将输入的百分制成绩按以下规定转换成相应的等级：

成 绩	等级
100~90	优秀
89~80	良好
79~70	中等
69~60	及格

59~0

不及格

```
#include <iostream>
using namespace std;
int main()
{
    float score;
    cout<<"请输入成绩:";
    cin>>score;
    if(score>=0 && score<=100)
        switch(__1__)
        {
            case 10:
            case 9: cout<<score<<"分：优秀"<<endl;    break;
            case 8: cout<<score<<"分：良好"<<endl;
break;
            case 7: cout<<score<<"分：中等"<<endl;
break;
            case 6: cout<<score<<"分：及格"<<endl;
break;

            __2__:cout<<"分：不及格"<<endl;
        }
    else
        cout<<"输入数据有误!"<<endl;
```

```
        return 0;
    }
}
```

参考答案 1: `int(score)/10` 2: `default`

输入两个整数a和b，按从小到大的顺序输出这两个数。

```
#include <iostream>

using namespace std;

int main()
{
    int a, b, t;
    cout<<"请输入两个整数a,b:"<<endl;
    cin>>a>>b;
    if(a>b)
    {__1__;a=b;__2__;}
    cout<

    return 0;
}
```

参考答案 1: `t=a` 2: `b=t`

根据x的值，计算分段函数y的值。y的计算公式为： $y=|x|$
($x<0$), $y=e^x \sin x$ ($x \leq 0 < 10$), $y=x^3$ ($10 \leq x < 20$), $y=(3+2x) \ln x$ ($x \geq 20$)

```
#include <iostream>

#include <cmath>

using namespace std;

int main()
```

```

{
    double x,y;
    cout<<"请输入x: ";
    cin>>x;
    if(x<0)
        y=fabs(x);
    else if(__1__ )
        y=exp(x)*sin(x);
    else if(x<20)
        y=pow(x,3);
    else
        y=__2__(3+2*x);
    cout<<"y="<<y<<endl;
    return 0;
}

```

参考答案 1: $x < 10$ 或 $10 > x$ 2: $\log(x) * 2$

有20只猴子吃掉50个桃子。已知公猴每只吃5个，母猴每只吃4个，小猴每只吃2个。求出公猴、母猴和小猴各多少只。

```

#include <iostream>
using namespace std;
int main()
{
    int a,b,c;

```

```

        for(a=1;a<=11;a++)
            for(b=1;b<=13;b++)
            {
                c= __1__
;
                if(__2__
==50)
                    cout<<"公猴="<<a<<"，母猴="<<b<<"，
小猴="<<c<<endl;
            }
return 0;
}

```

参考答案 1: $20-a-b$ 2: $5*a+4*b+2*c$

求1000内所有的完数。所谓“完数”是指与其因子之和相等的数（除本身之外）。例如： $6=1+2+3$ ，而1、2和3都是6的因子。要求以如下形式输出：6——>1, 2, 3。

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i, j, sum;
```

```
    for(i=2;i<=1000;i++)
```

```
    {
```

```
        for(sum=1, j=2; j<=i/2; j++)
```

```
            //求i的因
```

子和

```
        if(i%j==0) sum+=j;
    if( __1__ ) //判断i是否
为完数
    {
        cout<<i<<"-->1";
        for(j=2;j<=i/2;j++) //按指定
格式输出完数
            if(i%j==0) cout<<', '<< __2__ ;
        cout<<endl;
    }
}
return 0;
}
```

参考答案 1: sum==i 或 i==sum 2: j

计算 pi 的近似值，直到最后一项的绝对值小于 10^{-8} 为止，近似公式为

```
{
le-8)
}
```

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

```

int main()
{
    double sum=0, faction=1;
    int denominator=1;
    int sign=1;
    while(fabs(faction)>1e-10)
    {
        faction=faction*_2_
+=faction;
        denominator+=2;
        sign*=-1;
        faction=sign/double(denominator);
    }
    sum*=4;
    cout<<" $\pi \approx$ "<<setiosflags(ios::fixed)<<setprecision(8)
<<sum<<endl;
    return 0;
}

```

参考答案 1: >= 2: sum

从键盘上输入一串字符，以Ctrl+Z (^Z) 表示输入结束。统计其中包含的单词的个数、字母的 个数、数字个数。规定单词之间用一个空白符分开（空白符包括空格符、水平制表符、换行 符）。

```

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

```

```

char c;
int alpha(0), num(0), ch(0), word(0);
while((c=getchar())!=EOF)
    // 使用getchar()函数逐一读取字符；EOF代表文本结束符，键
    盘上对应输入Ctrl+Z
    {
        if(c==' ' || c=='\t' || c=='\n')
            word++;
        if(c>='a' && c<='z' || c>='A' && c<='Z')
            alpha++;
        else if(__1__ )
            num++;
        else __2__ ;
    }

    cout<<"字母数="<<alpha<<" 数字数="<<num<<" 其他字符数
    ="<<ch ;

    cout<<" 单词数="<<word<<endl ;

    return 0;
}

```

参考答案 1: (int)c>=48 && (int)c<=57或c>='0' && c<='9' 或c>=48 && c<=57 2: ++ch或ch++或ch=ch+1或ch+=1

计算由0到9之间的任意3个不相同的数字组成的三位数共有多少种不同的组合方式。

```
#include <iostream>
```



```

using namespace std;
int main( )
{
    int i, j, k, count=0;
    for ( i=9; i>=1; i--)
        for ( j=9; j>=0; j--)
            if ( __1__ )continue;
            else
                for(k=0;k<=9;k++)
                    if(__2__ )    count++;

    cout<<count<<endl;
    return 0;
}

```

参考答案 1: $i==j$ 或 $j==i$ 2: $(k!=i)\&\&(k!=j)$ 或 $k!=i \ \&\& \ k!=j$ 或 $k!=j \ \&\& \ k!=i$

“同构数”是指这样的数：它恰好出现在平方数的右端。

例如： $376*376=141376$

试找出10000以内的全部同构数。

```

#include <iostream>
using namespace std;
int main()
{
    int n, sqr;

```

```

for (n=1;n<10000;n++)
{
    if (n<10)
        sqr=__1__
;
    else
        if (n<100)
            sqr=n*n%100;
        else
            if (n<1000)
                sqr=n*n%1000;
            else
                sqr=n*n%10000;
        if (__2__
)
        cout<<n<<' *' <<n<<' =' <<n*n<<endl;
    }
    return 0;
}

```

参考答案 1: $n*n\%10$ 2: $sqr==n$

计算1000以内能被3整除的自然数之和。

```
#include <iostream>
```

```
using namespace std;
```

```

int main( )
{
    int x=1,sum=0;
    while(true)
    {
        if ( __1__ ) break;

        if ( __2__ ) sum+=x;
        x++;
    }
    cout<<sum<<endl;
    return 0;
}

```

参考答案 1: $x > 1000$ 或 $1000 < x$ 2: $x \% 3 == 0$ 或 $0 == x \% 3$ 或 $x == x / 3 * 3$

Fibonacci数列的头两个数分别是 0 和 1，从第三个数开始，每个数等于前两个数的和。求 Fibonacci数列的前20个数。要求每行输出5个数。

```

#include <stdio.h>
#include <iomanip>
using namespace std;
int main( )
{
    int f, f1, f2, i;
    cout<<"Fibonacci数列: \n";

```

```

    f1=0;f2=1;
    cout<<setw(6)<<f1<<setw(6)<<f2;
    for(i=3;i<=20;i++)
    {   f=__1__;
        cout<< f;
        f1=f2; f2=__2__
;
    }
    cout<<endl;
    return 0;
}

```

参考答案 1: f1+f2 或 f2+f1 2: f

程序设计语言的基本单词有标识符、__1__和运算符。

参考答案 1: 关键字

与分支语句有关的关键字有if、__1__、switch和case。

参考答案 1: else

C++源程序中分隔符有空格符、换行符、水平制表符、__1__、；和：。

参考答案 1: ，

循环语句使用的关键字是__1__、__2__和 __3__。

参考答案 1: do 2: while 3: for

注释语句不是__1__，单行注释以__2__开始；多行注释语句以/*开始，以__3__结束。

参考答案 1: 可执行语句 2: // 3: */

程序的实质是__1__和算法。

参考答案 1: 数据

C++源程序必须经过__1__和__2__才可作为执行程序。

参考答案 1: 编译 2: 连接

cout<<__1__<<120语句可以用8进制数（输出）表示120 。

参考答案 1: oct

若do循环的“尾”为while (i++<10)”，并且i的初值为0，同时在循环体中不会修改i的值，则 循环体将被重复执行__1__次后正常结束。

参考答案 1: 10

当在程序中执行到__1__语句时，将结束本层循环类语句或switch语句的执行。

参考答案 1: break

当在程序中执行到__1__语句时，将结束所在循环语句中循环体的一次执行。

参考答案 1: continue

程序运行中需要从键盘上输入多个数据时，各项数据之间应使用__1__符号作为分隔符。

参考答案 1: 空格或回车

若while循环的“头”为while (i++<=10)”，并且i的初值为0，同时在循环体中不会修改i的值，则 循环体将被重复执行__1__次后正常结束。

参考答案 1: 11

1、在代码段for(int i=0;i<=10;i++)

{循环体}

中循环体共执行：__1__次。

参考答案 1： 11

只有一个分号的语句叫：__1__。

参考答案 1： 空语句

用{}括起来的语句叫：__1__。

参考答案 1： 复合语句

设int a=15,b=26;则cout <<(a,b);的输出结果是：__1__。

参考答案 1： 26

有以下定义：

```
int x=10,y=20,z=30;
```

执行下列语句后，x的值是

__1__，y 的值是__2__，z 的值是__3__。

参考答案 1： 20 2： 30 3： 30

以下程序的功能为：输出一个整数中各位数中的最大值，如输入为12935，则输出结果为9。填 空补充程序。

```
#include <iostream>

using namespace std;

int main()
{
    int x,max=0;
    cin>>x;
    while(x>0)
    {
        if(
```

```

__1__>max)      max=__2__;
x/=10;
}

cout<<max<<endl;
return 0;
}

```

参考答案 1: x%10 2: x%10

根据以下嵌套的if语句所给的条件，填写 switch 语句，使它完成相同的功能。设 k、mark 均为 int 变量，且mark 的取值在1~100之间。填空补充程序中的语句。

if语句：

```

if(mark<60)  k=1;
else if(mark<70)  k=2;
else if(mark<80)  k=3;
else if(mark<90) k=4;
else if(mark<=100) k=5;

```

switch 语句：

```

switch
__1__ {__2__: k=1;break;
case 6: k=2; break;
case 7: k=3; break;
case 8: k=4;  break;
__3__: k=5;
}

```

参考答案 1: (mark/10) 2: case 0:case 1:case 2: case 3: case 4:case 5 3: case 9:case 10:

有说明语句: `int x=10, y=10;` 执行语句

`cout<<setw(2)<<hex<<(x%y, x-y, x+y);`

输出结果为

__1__。

参考答案 1: 14

`while(!x!=0) {...}` 循环结构中的循环条件的含义是__1__。

参考答案 1: `x==0`

有下列程序:

```
#include<iostream>
using namespace std;
main() {
    int y=9;
    for(;y>0;y--)
        if(y%3==0)
            cout<<(--y);
}
```

程序运行的结果是:

__1__。

参考答案 1: 8 5 2

下列程序用于判断a、b、c能否构成三角形,若能,输出yes,否则输出no。确定a、b、c能构成三角形的条件是需同时满足三个条件:
 $a+b>c$, $a+c>b$, $b+c>a$ 。请填空。


```

#include<iostream>
using namespace std;
main() {
float a,b,c;
cin>>a>>b>>c;
if (
__1__)
cout<<" yes" <<endl;
else
cout<< "no " <<endl;
}

```

参考答案 1: $(a+b>c)\&\&(a+c>b)\&\&(b+c>a)$

下面程序的功能是：输出100以内能被4整除且个位数为8的所有整数，请填空。

```

#include<iostream>
using namespace std;
main()
{ int i,j;
for(i=0;
__1__ ;i++)
{j=i*10+8;
if (j%4!=0 continue;
cout<<j<< " ";

```

```
}
```

参考答案 1: i<10;

下面程序的功能是：计算1~20之间的奇数之和与偶数之和，请填空使程序功能完整。

```
#include<iostream>

using namespace std;

main() {
    int a=0, b=0, c=0, i;
    for(i=0; i<=20; i+=2) {
        a+=i;
        __1__;
        c+=b;
    }
    cout<<" 偶数之和=" <<a;
    cout<<" 奇数之和=" <<(c-21);
}
```

参考答案 1: b=i+1

下面程序段中的变量已经正确定义：

```
for(i=0; i<4; i++, i++)
for(k=1; k<3; k++);
cout<<" *";
```

该程序段的输出结果是

__1__。

参考答案 1: *

以下程序的功能是：从键盘上输入若干学生的成绩，统计并输出最高成绩和最低成绩，当输入 负数时，结束输入，请填空。

```
#include<iostream>

using namespace std;

void main() {

float x, aMax, aMin;

cin>>x;

aMax=x;

aMin=x;

while(

__1__) {

if (x>aMax)

aMax=x;

else if (

__2__)

aMin=x;

cin>>x;

}

cout<<" max=" <<aMax<< "min "<<aMin;

}
```

参考答案 1: $x > 0$ 2: $x < aMin$

下面程序的执行结果是__1__

```
#include<iostream>
```

```
using namespace std;
main() {
    int j;
    for(j=10;j>3;j--) {
        if (j%3)
            cout<<j;}
    }
```

参考答案

1: 96

已知“int x=10,y=20,z=30”，以下语句执行后，y的值是
 __1__if(x>y) z=x; x=y; y=z+1;

参考答案 1: 31

执行以下程序段时，循环体将执行__1__次。

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

参考答案 1: 2

本程序想实现当x<y<z时，输出x<y<z：请填空：

```
if(
    __1__)
    cout<<x<<" <" <<y<<" <" <<z<<endl;
```

参考答案 1: x<y&&y<z

若有定义 “int a=7;float x=2.5,y=4.7;”, 则表达式 $x+a\%3*(int)(x+y)\%2/4$ 的值是__1__。

参考答案 1: 2.500000

设有以下语句:

```
char x=3, y=6, z;
```

```
z=x^y<<2;
```

则z的二进制值是__1__。

参考答案 1: 00011011

下列语句段的输出结果是__1__

```
int k=1, n=10, m=11;
```

```
while(k<=n) {
```

```
    m*=2;
```

```
    n--;
```

```
}
```

```
cout<<m<<endl;
```

参考答案 1: 1024

结构化程序由三种基本结构组成, 即: 顺序结构、__1__结构、循环结构.

参考答案 1: 选择 或 分支

switch语句中, 关键字switch后面的表达式可以是__1__型、字符型或枚举型表达式

参考答案 1: 整 或 int

```
int a=1, b=2, c=3, d;
```

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

```
else d=b;
```

```
if(c>d) d=c;
```

运行完该程序段后, d的值是

__1__。

参考答案 1: 3

设x和y均为整型变量, 且x=3, y=2, 则1.0*x/y表达式的值为

__1__。

参考答案 1: 1.5

阅读程序, 严格按输出顺序与格式写出程序运行结果

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

```
int s=0,i=0;
```

```
main()
```

```
{ while (++i<=4)
```

```
{ if (i==3) continue;
```

```
switch(i)
```

```
{ case 1:s+=i;break;
```

```
case 2: s+=i;
```

```
case 3:s+=i;break;
```

```
default:s+=i;
```

```
}
```

```
}
```

```
cout<<s;
```

```
}
```

参考答案 9

阅读程序，严格按输出顺序与格式写出程序运行结果

```
#include <iostream.h>
#include <iomanip.h>
main()
{ int i,k;
  for (i=0;i<6;i++)
    {for(k=1;k<=i;k++)
      cout<<setfill(' ')<<" ";
    cout<<setw(6-i)<<setfill('*')<<" "<<endl;}
}
```

参考答案

**

*

如下程序的运行结果：

```
#include <iostream>
using namespace std;
int main()
{ int n;
  for(n=1;n<6;n++)
```

```

        {if(n%2)   cout<<"1";
          else   continue;
        cout<<"2";}
        cout<<endl;
        return 0;
    }

```

参考答案 1212

```

#include <iostream>
using namespace std;
int main()
{unsigned num=13456, y=0, digital;
  do {
    digital=num%10;
    num/=10;
    y=y*10+digital;
  }while(num>0);
  cout<<"转换后的数据为: "<<y<<endl;
  return 0;}

```

参考答案 转换后的数据为: 65431

```

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

```



```

{
for(int i=1;i<=30;i++)
{
    if(i%7!=0)
        continue;
    cout<<setw(5)<<i;
}
cout<<endl;
    return 0;
}

```

参考答案 7 14 21 28

写出运算结果:

```

#include <iostream.h>
void main()
{   int a,b,c,s;
    a=9;b=4;c=2;
    s=++a%b<a*c?b+++1:a && b;
    cout<<s;
}

```

参考答案 5

写出运算结果:

```

#include <iostream.h>
void main()

```

```

{   int  a,b,c,s;

    a=2;b=2;c=11;

    s=a>c%b && a*c<c+10;

    cout<<s;

}

```

参考答案 0

写出运算结果：

```

#include <iostream.h>

void main()

{   int  a,b,c,s;

    a=10;b=2;c=11

    s=++c+200>a*b+c%b || a && (c*10+b, b=300);

    cout<<s;

}

```

参考答案 1

写出运算结果：

```

#include <iostream.h>

void main()

{   int  s;

    s=sizeof(float)

    cout<<s;

}

```

参考答案 4

写出运算结果：

```
#include <iostream.h>

void main()
{   char a;
    double b;
    int c,s;
    a=102;b=212.41;c=1100
    s=sizeof(a/b+c);
    cout<<s;
}
```

参考答案 8

写出运算结果：

```
#include <iostream.h>

void main()
{   int a,b,c,s;
    a=11;b=21;c=41;
    s=(++a/b,b/c);
    cout<<s<<" "<<a;
}
```

参考答案 0, 12

写出运算结果：

```
#include <iostream.h>

void main()
```

```

{   int  a, b, c, s;
    a=10;b=2;c=11;
    s=a>b, a+c, c*10;
    cout<<s;
}

```

参考答案 1

写出运算结果：

```

#include <iostream.h>
void main()
{   int  a, b, c, s;
    a=100;b=21;c=5;
    s=a<b?(a<c?a:c):b<c?b:c;
    cout<<s;
}

```

参考答案 5

```

#include <iostream.h>
void main()
{   int  a, b, c, s;
    a=20;b=2;c=11;
    s=a+(s=(a+c)%2+b);
    cout<<s;
}

```

参考答案 23

写出运算结果：

```
#include <iostream.h>

void main()
{   int a,b,c,s;
    a=100;b=20;c=110
    s=a<b?a:c;
    cout<<s;
}
```

参考答案 110

写出运算结果：

```
#include <iostream.h>

void main()
{   int a,b,c;
    a=123;b=10;c=100;
    cout<<(a-a/b*10)<<" "<<(a/b-a/c*10);
}
```

参考答案 3,2

写出运算结果：

```
#include <iostream.h>

void main()
{   float a,s;
    a=456.237;
    s=(int)(a*100+0.5)/100.0;
```

```
    cout<<s;
}
```

参考答案 456.24

写出运算结果：

```
#include <iostream.h>

void main()
{   int a,b,s=10;
    a=10;b=2;
    s*=a && b;
    cout<<s;
}
```

参考答案 10

写出运算结果：

```
#include <iostream.h>

void main()
{   int a,b,s=20;
    a=10;b=1;
    s/=a || b--;
    cout<<s<<" "<<b;
}
```

参考答案 20, 1

写出运算结果：

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

```

void main()
{
    char a,s;

    a=0xfe;

    s=~a;

    cout<<s;

}

```

参考答案 1

写出运算结果：

```

#include <iostream.h>

void main()
{
    char a,b,s;

    a=0xfe;b=0x02;

    s=a & b;

    cout<<(int) s;

}

```

参考答案 2

写出运算结果：

```

#include <iostream.h>

void main()
{
    char a,b,s;

    a=0x01;b=0x02;

    s=a | b;

    cout<<(int) s;

}

```

```
}
```

参考答案 3

写出运算结果:

```
#include <iostream.h>

void main()

{   int a,b,c,s;

    a=10;b=2;c=11;

    s=c<a/b;

    cout<<s;

}
```

参考答案 0

写出运算结果:

```
#include <iostream.h>

void main()

{   int a,b,c,s;

    a=10;b=0;c=11;

    s=a && b || !c;

    cout<<s;

}
```

参考答案 0

写出运算结果:

```
#include <iostream.h>

void main()
```



```

{   int  a,b,c,s;

    a=10;b=2;c=11;

    s=(a+c)%3 && b--;

    cout<<s<<" "<<b;

}

```

参考答案 0, 2

写出运算结果:

```

#include <iostream.h>

void main()

{   int  a,b,c;

    float s;

    a=10;b=20;c=11;

    s=(float) a/b+c;

    cout<<s;

}

```

参考答案 11.5

写出运算结果:

```

#include <iostream.h>

void main()

{   int  a,b,c,s;

    a=10;b=2;c=11;

    s=a+b>c;

    cout<<s;

}

```

```
}
```

参考答案 1

写出运算结果：

```
#include <iostream.h>
void main()
{   int a,b,c;
    float s;
    a=10;b=2;c=5;
    s=b/a+1/b+1/c;
    cout<<s;
}
```

参考答案 0

写出运算结果：

```
#include <iostream.h>
void main()
{   int a,b,c;
    float s
    a=10;b=2;c=2;
    s=1.0/b+1.0*a/c;
    cout<<s;
}
```

参考答案 5.5

写出运算结果：

```

#include <iostream.h>

void main()
{
    int a,b,c,s;
    a=10;b=3;c=11;
    s=(a--+c)%b+2;
    cout<<s<<" "<<a;
}

```

参考答案 2, 9

写出运算结果:

```

#include <iostream.h>

void main()
{
    int a,b,c,s;
    a=10;b=6;c=11;
    s=(++a+c)%b+10;
    cout<<s<<" "<<a;
}

```

参考答案 14, 11

写出运算结果:

```

#include <iostream.h>

void main()
{
    int a,b,c,s;
    a=10;b=2;c=11;
    s=c+a/2*b;
}

```

```
    cout<<s;
}
```

参考答案 21

写出运算结果：

```
#include <iostream.h>

void main()
{   int a,b,c,s;
    a=10;b=2;c=11;
    s=(a+c)%2+b;
    cout<<s;
}
```

参考答案 3

```
#include <iostream>
using namespace std;
int main()
{
    for(int a=1,b=1;a<=6;a++)
    {   if(b>=10) break;
        if(b%2==1){b+=7;continue;}
        b-=5;
    }
    cout<<"a="<<a<<', '<<"b="<<b<<endl;
    return 0;
}
```

```
}
```

参考答案 a=4, b=10

```
#include <iostream>using namespace std;int main() {    int  
k=0;char c='A';    do { switch(c++) { case 'A':  
k++;break;    case 'B': k--;    case 'C':  
k+=2;break;    case 'D': k%=2;continue;    case  
'E': k*=20;break;    default: k/=5; } k++; }while  
(c<'G'); cout<<"k="<<k<<endl; return 0;}
```

参考答案 k=5

```
#include <iostream>  
  
#include <iomanip>  
  
using namespace std;  
  
const int N=10;  
  
int main()  
{int i=1;  
  for(;i<=N;i++)  
  {  
      if(i%5!=0)  
          continue;  
      cout<<setw(5)<<i;  
  }  
  
  cout<<endl;  
  
  return 0;  
}
```

参考答案 5 10

```

#include <iostream>
using namespace std;
int main()
{unsigned num(895),digital,result(0);
do
{
    digital=num%10;
    num/=10;
    result=result*10+digital;
}while(num>0);
cout<<"result="<<result<<endl;
return 0;
}

```

参考答案 result=598

```

#include <iostream>
using namespace std;
int main()
{int i(1),a(0);
for (;i<=5;i++)
{
    do{
        i++;
        a++;
    }
}
}

```

```

        }while(i<3);
        i++;
    }
    cout<<a<<" "<<i<<endl;
    return 0;
}

```

参考答案 3,8

```

#include <iostream>
using namespace std;
int main()
{ int a(5),b(30),k;
  k=a*b;
  while (a!=b)
      { if (a>b) a-=b;
        if (a<b)b-=a;
      }
  k/=a;
  cout<<"k="<<k<<endl;
  return 0;
}

```

参考答案 k=30

```

#include <iostream>
using namespace std;

```

```

int main()
{int n(6),k;
cout<<"结果是: "
for(k=1;k<=n;k++)
    if(n%k==0)    cout<<k<<'  ' ;
cout<<endl;
return 0;
}

```

参考答案 结果是: 1 2 3 6

```

#include <iostream>
using namespace std;
int main()
{
int a,b,c=0;
for(a=1;a<4;a++)
    for(b=4;b>1;b--)
    {
        if((a+b)%3==2){c+=a+b;cout<<a<<' ' <<b<<',' ;}
        if(c>20)break;
    }

    cout<<"c="<<c<<endl;
return 0;
}

```



```
}
```

参考答案 1 4, 2 3, 3 2, c=15

```
#include <iostream>
```

```
using namespace std;
```

```
const int T=5;
```

```
int main()
```

```
{
```

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

```
    for(i=1; i<=T; i+=2)
```

```
        for(j=2; j<=T; j++)
```

```
            if(i+j==T) cout<<'+' ;
```

```
            else if(i*j==T) cout<<'*' ;
```

```
            else k++;
```

```
    cout<<"k="<<k<<endl;
```

```
    return 0;
```

```
}
```

参考答案 +*k=4

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int m(64), n(16);
```

```
    while(m!=n)
```

```

        while(m>n) m-=n;
        while(n>m) n-=m;
    }

    cout<<"m="<<m<<endl;
    return 0;
}

```

参考答案 m=16

```

#include <iostream>
using namespace std;
int main()
{
    int k=0;char c='A';
    do
    { switch(c++)
        { case 'A': k++;break;
          case 'B': k--;
          case 'C': k+=2;break;
          case 'D': k%=2;continue;
          case 'E': k*=20;break;
          default: k/=5;
        }
        k++;
    }while(c<'G');
}

```

```
    cout<<"k="<<k<<endl;
    return 0;
}
```

参考答案 k=5

```
#include <iostream>
using namespace std;
int main()
{
    for(int a=1,b=1;a<=6;a++)
    { if(b>=10)break;
      if(b%2==1){b+=7;continue;}
      b-=5;
    }
    cout<<"a="<<a<<', '<<"b="<<b<<endl;
    return 0;
}
```

参考答案 a=4, b=1

计算斐波那契数列第n项的函数定义如下：

```
int fib(int n)
{
    if (n==0)    return 1;
    else if (n==1) return 2;
                else return fib(n-1)+fib(n-2);
}
```

}

若执行函数调用表达式fib(2)，函数fib被调用的次数是()。

A 1 B 2 C 3 D 4 参考答案 B

若各选项中所用变量已正确定义，函数fun中通过return语句返回一个函数值，以下选项中错误的程序是：

A main() {……x=fun(2,10);……} float fun(int a,int b)
{……}

B float fun(int a,int b){……} main() {……x=fun(i,j);……}

C float fun(int,int); main() {……x=fun(2,10);……} float
fun(int a,int b) {……}

D main() { float fun(int i,int j); ……x=fun(i,j);…… } float
fun(int a,int b) {……} 参考答案 A

有以下程序： #include<iostream.h> char fun(char x , char y) {
if(x<y) return x; return y; } void main() { int
a='9',b='8',c='7'; cout<<fun(fun(a,b),fun(b,c))<<endl; } 程
序的执行结果是_____。

A 函数调用出错 B 8 C 9 D 7 参考答案 D

下述关于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 D 函数不能自己调用自己 参考答案 C

有如下程序： #include<iostream> using namespace std; int main(
) { void function(double val); double val;
function(val); cout<<val; return 0; } void
function(double val) { val=3; } 编译运行该程序将出现的情况
是()。

A 编译出错，无法运行 B 输出：3 C 输出：3.0 D 输出一个不确定的
数 参考答案 D

有下列程序： #include<iostream.h> fun(int a, int b) { int

```
c;    c=a+b;    return c; } void main( ) {    int
x=6, y=7, z=8, r;    r=fun((x--, y++, x+y), z--);    cout<<r<<endl;
} 则该程序执行结果是 (    )。
```

A 11 B 20 C 21 D 31 参考答案 C

```
下列函数的运行结果是 (    )。 #include<iostream.h> int
add(int a,  int b); void main( ) {    extern int x,y;
cout<<add(x,y)<<endl; } int x(20),y(5); int add(int a,  int b)
{    int s=a+b;    return s; }
```

A 25 B 30 C 20 D 15 参考答案 A

```
下列程序运行后的输出结果是 (    )。 #include<iostream.h>
void fun(int,  int,  int *); void main( ) {    int x,  y,  z;
    fun(5,  6,  &x);    fun(7,  x,  &y);    fun(x,  y,  &z);
cout<<x<<" "<<y<<" "<<z<<endl; } void fun(int a, int b, int
*c) {    b+=a;    *c=b-a; }
```

A 5,5,5 B 6,6,6 C 5,6,7 D 7,7,7 参考答案 B

```
下列程序输出的结果是 (    )。 #include<iostream.h> void
fun1(char a, char b){char c; c=a; a=b; b=c;} void
fun2(char*a, char b){char c;c=*a; *a=b; b=c;} void
fun3(char*a, char*b){char c;c=*a; *a=*b; *b=c;} void main( ) {
    char a,b;    a='A';b='B'; fun1(a,b); cout<<a<<b;
a='A';b='B'; fun2(&a,b); cout<<a<<b;
a='A';b='B'; fun3(&a,&b); cout<<a<<b;    cout<<'\\n'; }
```

A BABBAB B ABBBBBA C ABBABA D ABABBA 参考答案 B

在C++程序中，如果要求通过函数来实现一种简单的功能，并且要求尽可能加快程序的执行速度，则应该选用 ()。

A 内联函数 B 重载函数 C 递归调用 D嵌套调用 参考答案 A

```
下列程序的运行结果是 (    )。 #include<iostream.h> void
fun(int &a, int b=3) {    static int i=2;    a=a+b+i;
i=i+a; } void main( ) {    int x=5,y=2;    fun(x, y);
cout<<x<<" ";    fun(x);    cout<<x<<endl; }
```

A 5, 2 B 7, 11 C 11, 23 D 9, 23 参考答案 D

若有以下变量定义和函数调用语句: `int a=5; fun(&a);` 则执行下面函数后正确的输出结果是 ()。 `void fun(int *x) { cout<<++*x<<endl; }`

A 3 B 4 C 5 D 6 参考答案 D

对下列程序段的描述正确的是 ()。 `#include<iostream.h> int fun(int, int); void main() { cout<<fun(1,2)<<endl; } int fun(int x, int y) { return x+y; }`

A 该函数定义正确, 但函数调用方式错误 B 该函数调用方式正确, 但函数定义错误 C 该函数定义和调用方式都正确 D 该函数定义和调用方式都错误 参考答案 C

下列函数中, 对调用它的函数没有起到任何作用的是 ()。

A `void f1(double &x){ --x; }` B `double f2(double x){ return x-1.5; }` C `void f3(double x){ --x; }` D `double f4(double *x){ --*x; return *x; }` 参考答案 C

下列程序的输出结果是 ()。 `#include<iostream.h> void fun(int a, int b) { int temp; temp=a; a=b; b=temp; } void main() { int m=1, n=2; fun(m, n); cout<<m<<n<<endl; }` A 12 B 21 C 22 D 程序有错误 参考答案 A

以下说法中, 正确的是_____。

A 在C++的程序中, 要调用的函数必须在main()定义 B C++程序总是从最先定义的函数开始执行 C C++程序中的main()函数必须放在程序的开头部分 D C++程序总是从main()函数开始执行 参考答案 D

决定C++语言中函数的返回值类型的是_____。

A return语句中的表达式类型 B 调用该函数时系统随机产生的类型 C 调用该函数时的主调用函数类型 D 在定义该函数时所指定的数据类型 参考答案 D

以下程序的输出结果是_____。

```

#include<iostream.h>

void main()

{   int x=1,y=3;
cout<<x++<<"， ";
{int x=0;x+=y*2;
    cout<<x<<"， "<<y<<"， ";  }
cout<<x<<"， "<<y;
}

```

A 1, 6, 3, 1, 3 B 1, 6, 3, 6, 3 C 1, 6, 3, 2, 3 D 1, 7, 3, 2, 3 参考答案 C

以下有关函数存储类别的说明不正确的是_____。

A 静态函数用static定义 B 必须加extern关键字定义外部函数 C 内部函数不能在定义文件外调用 D 定义文件外调用外部函数，必须用extern声明 参考答案 B

```

#include <iostream.h>

#define zw 0

main( )
{
    int a;
    #if zw
        cout<< "输入整数： " ;
    #else
        cout<< "input integer: " ;
    #endif
}

```

```
        cin>>a;
    }
```

以上程序运行时将提示_____。

A 空白 B 中文 C 先中文后英文 D 英文 参考答案 D

有关内联函数不正确的说法是_____。

A 可加快程序的运行速度 B 可缩短执行代码 C 增强源程序的可读性
D 增强源程序的可维护性 参考答案 B

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

```
int sp(int a=1)
{
    static int b;
    return ++b+a;
}
```

```
void main( )
{
    sp( );
    cout<<sp( );
}
```

以上程序输出值为_____。

A 4 B 1 C 3 D 2 参考答案 C

与函数无关的是_____。

A 参数宏 B 函数名 C 参数类型 D 返回值 参考答案 A

使用sin(x), 必须包括的头文件为_____。 A iostream.h

B math.h C stdlib.h D string.h 参考答案 B

函数调用语句“f((e1, e2), (e3, e4));”中参数的个数为：_____。

A 1 B 2 C 3 D 4 参考答案 B

以下语句的输出结果是_____

```
enum weekday {sun, mon=3, tue, wed, thu};
```

```
weekday    workday;
```

```
workday=wed;
```

```
cout<<workday<<endl;
```

A 3 B 5 C 1 D 4 参考答案 B

在函数内部或复合语句内部定义的变量称为_____。

A 内部变量 B 外部变量 C 局部变量 D 全局变量 参考答案 C

在C++中，当用cin、cout流对象输入、输出时，程序头部要包括()。

A #include<stdio.h> B #include<iostream.h>

C #include<stdlib.h> D #include<string.h> 参考答案 B

以下说法中正确的是_____。 A C++程序总是从第一个的函数开始执行
B 在C++程序中, 要调用的函数必须在main()函数中定义 C C++程序总是从main()函数开始执行
D C++程序中的main()函数必须放在程序的开始部分 参考答案 C

以下叙述中不正确的是_____。

A 在不同的函数中可以使用相同名字的变量 B 函数中的形式参数是局部变量
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 参考答案 D

在C++程序中，以下说法正确的是_____。

A 函数的定义可以嵌套, 但函数的调用不可以嵌套 B 函数的定义不可以嵌套, 但函数的调用可以嵌套
C 函数的定义和函数的调用都不可以嵌套 D 函数的定义和函数的调用都可以嵌套 参考答案 B

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

```

void fun(int x, int y, int z)
{
    z=x*x+y*y;
}
void main( )
{
    int a=31;
    fun(5, 2, a);
    cout<<a;
}

```

以上程序的输出结果是_____。

A 0 B 29 C 31 D 无定值 参考答案 C

以下程序的输出结果是()。

```

#include <iostream.h>
void fun(int k)
{
    if (k>0) fun(k-1);
    cout<<k;
}
void main( )
{
    int w=5;
    fun(w);
    cout<<endl;
}

```

A 54321 B 012345 C 12345 D 543210 参考答案 B

以下所列的各函数首部中，正确的是_____。

A void play(var :Integer, var b:Integer) B void play(int a,b)
C void play(int a,int b) D Sub play(a as integer,b as integer) 参考答案 C

若有以下程序

```
#include <iostream.h>

long fun ( int n )
{
    long s;
    if (n==1 || n==2) s=2;
    else s=n-fun(n-1);
    return s;
}

void main( )
{ cout<<fun(3); }
```

则程序的输出结果是_____。

A 1 B 2 C 3 D 4 参考答案 A

若已定义的函数有返回值，则以下关于该函数调用的叙述中错误的是_____。

A 调用可以作为独立的语句存在 B 调用可以作为一个函数的实参 C 调用可以出现在表达式中 D 调用可以作为一个函数的形参 参考答案 D

有以下函数定义：

```
void fun( int n, double x) { ..... }
```

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_____。 A fun(int y,double m); B k=fun(10,12.5);

C fun(x,n); D void fun(n,x); 参考答案 C

在一个C++程序中，_____。

A main函数必须出现在所有函数之前 B main函数可以在任何地方出现
C main函数必须出现在所有函数之后 D main函数必须出现在固定位置
参考答案 B

有以下程序

```
#include<iostream.h>

char fun(char x , char y)
{
    if (x<y) return x;
    else return y;
}

void main( )
{
    char a='9',b='8',c='7';
    cout<<fun(a,b)<<fun(b,c) ;
}
```

程序的执行结果是_____。

A 函数调用出错 B 78 C 97 D 87 参考答案 D

若程序中定义了以下函数

```
double myadd (double a,double B)
{ return (a+B) ;}
```

并将其放在调用语句之后，则在调用之前应该对该函数进行声明，以下选项中错误的说明是_____。

A double myadd(double a,B) ; B double myadd(double,double);
C double myadd(double b,double A) ; D double myadd(double
x,double y); 参考答案 A

有以下程序

```
#include<iostream.h>

#define P 3

#define F(int x) { return (P*x*x); }

void main( )
{
    cout<<F(3+5);
}
```

程序运行后的输出结果是_____。

A 192 B 29 C 25 D 编译出错 参考答案 D

以下关于函数的叙述中正确的是_____。

A 每个函数都可以被其它函数调用（包括main函数） B 每个函数都可以被单独编译 C 每个函数都可以单独运行 D 在一个函数内部可以定义另一个函数 参考答案 B

设函数fun的定义形式为 void fun(char ch, float x) { ...
}, 则以下对函数fun的调用语句中, 正确的是_____。

A fun("abc", 3.0); B t=fun('D', 16.5); C fun('65', 2.8);
D fun(32, 32); 参考答案 D

C++中规定, 程序中各函数之间_____。

A 既允许直接递归调用也允许间接递归调用 B 不允许直接递归调用也不允许间接递归调用 C 允许直接递归调用不允许间接递归调用 D 不允许直接递归调用允许间接递归调用 参考答案 A

建立函数的目的之一是_____。

A 提高程序的执行效率 B 提高程序的可读性 C 减少程序的篇幅 D .减少程序文件所占内存 参考答案 B

以下正确的函数定义形式是_____。

A double fun(int x,int y){} B double fun(int x; int y) {}
C double fun(int x, int y); {} D double fun(int x,y); {} 参考答案 A

简单变量做实参时，它和对应形参之间的数据传递方式为_____。 A 地址传递 B 单向值传递 C 由实参传给形参，再由形参传回给实参 D 由用户指定传递方式 参考答案 B

引入内联函数的目的是为了()。

A 节省空间 B 提高函数调用的效率 C 方便变量的使用 D 扩大局部变量的使用范围 参考答案 B

下列说法不正确的是_____。

A 函数必须有返回值 B 函数的返回值是通过return语句返回到主调函数的。 C return语句后的括号可要可不要 D 有函数值的函数，它返回的函数值的类型是确定的 参考答案 A

在函数调用过程中，如果函数funA调用了函数 funB,函数funB又调用了函数funA，则_____。

A 称为函数的直接递归调用 B 称为函数的间接递归调用 C 称为函数的循环调用 D C++中不允许这样的递归调用 参考答案 B

以下有关函数参数的叙述不正确的是_____。

A 函数的形参命名可以任意，只要符合标识符规则。 B 实参只能是常数 C 形参的值与实参的值不一定时刻保持一致。 D 函数参数的值也可以是内存单元地址 参考答案 B

函数原型是指下面哪种格式_____。

A 函数名（[参数1类型][,参数2类型][...]） B 函数类型关键字 函数名（[参数1类型][,参数2类型][...]）； C 函数类型关键字 函数名（标识符1[,标识符2][...]）； D 函数名（参数1类型,标识符1[,参数

2类型，标识符2][...]) ; 参考答案 B

下面说法不正确的是_____。

A C++程序就是由许多函数组成的。 B 一个程序必须有且只有一个主函数。 C 操作系统可以调用主函数，程序中的其他函数也可以调用主函数。 D 被调用的函数简称为被调函数。 参考答案 C

inline关键字的含义是_____。

A 外联 B 内联 C 宏定义 D 调用函数定义 参考答案 B

不能放在声明区的语句是_____。

A if语句 B 变量定义语句 C 函数声明语句 D 类定义语句 参考答案 A

可放在声明区的语句是_____。

A for语句 B while语句 C 结构体定义语句 D switch语句 参考答案 C

C++源程序可有多函数，但必须有一个_____。

A 初始函数 B 构造函数 C 析构函数 D 主函数即启动函数 参考答案 D

下列_____不写在声明区。

A 宏定义 B 包含命令 C 函数声明 D 函数定义 参考答案 D

C++中的函数定义都是平行的，这意味着（ ）。

A 在一个函数中可以定义另一个函数 B 在一个函数中不能调用另一个函数 C 在一个函数中可以定义另一个函数，但不能调用该函数 D 在一个函数中不能定义另一个函数，但可以调用其他标准函数或自定义函数
参考答案 D

一个函数的返回值是通过函数中的（ ）语句获得的。

A return B exit C break D value 参考答案 A

如果要定义一个没有返回值的函数，应使用的C++关键字是（ ）。

A NULL B empty C void D N/A 参考答案 C

声明动态存储变量的C++关键字是（ ）。

A dynamic B auto C static D register 参考答案 B

在声明一个变量的存储类别时，可以省略的关键字是（ ）。

A extern B auto C static D register 参考答案 B

有关内联函数不正确的说法是（ ）。

A 可加快程序的运行速度 B 可缩短执行代码 C 增强源程序的可读性
D 增强源程序的可维护性 参考答案 B

已知：int m=10；下列表示引用的方法中，正确的是（ ）。

A int &x=m; B int &x=10; C int &x; D float &t=&m 参考答案 A

通过函数来实现一种不太复杂的功能，并且要求加快执行速度，C++中采用（ ）。

A 重载函数 B 内联函数 C 递归函数 D 嵌套函数 参考答案 B

在C++中，关于下列设置参数默认的描述中，（ ）是正确的。

A 不允许设置参数的默认值 B 设置参数默认值只能在定义函数时设置
C 设置参数默认值时，应该是先设置左边的再设置右边的 D 设置参数默认值时，应该全部参数都设置 参考答案 B

当一个函数无返回值时，定义时函数的类型应该为（ ）。

A 任意 B void C int D 无 参考答案 B

考虑函数原型void test(int a, int b=7, char z=2)，下面的函数调用中，属于不合法调用的是。

A test(5) B test(5, 8) C test(5,6,7) D test() 参考答案 D

下列叙述中错误的是（ ）。

A 一个函数中可以有多条return语句 B 调用函数必须在一条独立的语句中完成
C 函数中通过return语句使函数值返回 D 主函数名main也可以带形参 参考答案 B

下列函数中，正确的递归函数是（ ）。

```
A int Fun(int n) {  
    if(n<1)  
        return 1;  
    else  
        return n*Fun(n+1);  
}
```

```
B  
int Fun(int n) {  
    if (abs(n)<1)  
        return 1;  
    else  
        return n*Fun(n/2);  
}
```

```
C  
int Fun(int n) {  
    if(n>1)  
        return 1;  
    else  
        return n*Fun(n*2);  
}
```

```
D  
int Fun(int n) {
```

```
if(n>1)
return 1;
else
return n*Fun(n-1);
}
```

参考答案 B

在不同文件中，对全局变量的引用要加上（）关键字。

A auto B register C static D extern 参考答案 D

下面的说明中正确的函数定义是()

A void fun(void) B void fun(void) {} C fun(int s) {} D fun(int s){return s+1;} 参考答案 D

若有以下函数调用语句” fun(a+b, (x, y), fun(n+k, d, (a, b))) ;”，在此函数中实参的个数是（）

A 3 B 4 C 5 D 6 参考答案 C

关于局部变量下列的说法正确的是（）

A 定义该变量的程序文件中的函数都可以访问。 B 定义该变量的函数中的定义处以下的任何语句都可以访问。 C 定义该变量的复合语句中的定义处以下的任何语句都可以访问。 D 局部变量可以用于函数中传递数据。 参考答案 C

有以下函数：

```
char fun(char *p){return p;}
```

则该函数的返回值是（）

A 无确切的值 B 形参p中存放的地址值 C 一个临时存储单元的地址 D 形参p自身的地址值 参考答案 B

以下对C++语言函数的描述中，正确的是

A C++程序由一个或一个以上的函数组成 B C++函数既可以嵌套定义又可以递归调用 C 函数必须有返回值，否则不能使用函数 D C++程序中调用关系的所有函数必须放在同一个程序文件中 参考答案 A

下列叙述中正确的是

A 全局变量的作用域一定比局部变量的作用域范围大 B 静态类别变量的生存期贯穿于整个程序的运行期间 C 函数的形参都属于全局变量 D 未在定义语句中赋初值的auto变量和static变量的初值都是随机值 参考答案 B

使用值传递方式将实参传给形参，下列说法正确的是

A 形参是实参的备份 B 实参是形参的备份 C 形参和实参是同一对象 D 形参和实参无联系 参考答案 A

C++语言中函数返回值的类型决定于

A return语句中的表达式类型 B 调用函数的主调函数类型 C 调用函数时变量的类型 D 定义函数时所指定的函数类型 参考答案 D

在下面的函数声明中，存在着语法错误的是

A void bc(int a, int) ; B void bc(int, int); C void bc (int, int x=5) ; D void bc(int x; int y); 参考答案 D

在函数中定义的变量及函数的形式参数，都属于

A 全局变量 B 局部变量 C 静态变量 D 常量 参考答案 B

在什么情况下，可以不用对自定义函数声明就可以调用（）

A 该被调用的自定义函数出现在同一文件的主调函数之前 B 主调函数在前，被调函数在后 C 该被调用函数出现在main函数之前 D 不存在这种情况 参考答案 A

设有一自定义函数的函数头形如：int f(int x, float y)。下列关于该函数的声明语句中，错误的是【 】。

A int f(int x, float y) B int f(int, float); C int f(int a, float b); D int f(int x, float z); 参考答案 A

说明函数类型时不需要指明每个函数参数的名字，只需要说明每个参数的类型和返回值类型就可以了。 [参考答案] 正确

如果一个函数没有形式参数，调用该函数时，可以省略括号对()。

[参考答案] 错误

在一个程序中，全局变量的名称与局部变量的名称不能相同。

[参考答案] 错误

内联函数可以递归调用。 [参考答案] 错误

在C++中，一个函数是一个编译单位。 [参考答案] 错误

main函数能调用其它函数，其它函数也能调用main函数

[参考答案] 错误

自定义函数必须有函数返回值 [参考答案] 错误

定义局部变量时使用static关键字的作用有两个：延长其生存期并扩大其作用域 [参考答案] 错误

下面是一个递归函数，其功能是使数组中的元素反序排列。请将函数补充完整。

```
void reverse(int *a, int size){  
    if(size<2) return;  
    int k=a[0];  
    a[0]=a[size-1];a[size-1]=k;  
    reverse(a+1, __1__); }  

```

参考答案 1: size-1

完成以下代码，实现求数列：

1, 3/2, 5/3, 8/5.....

前n项之和，n由主函数中给定。 #include <iostream.h> double

```
sum_s(int n); void main() { double s; int n; cout<<"
please input integer n:"; cin>>n; s=sum_s(n); cout<<"s="
<<s<<endl; } //sum_s函数求数列的前n项之和 double sum_s(int n)
{ int i; double t=1, s=1; for (i=1;i<n;i++) { __1__;
//递推关系t=(t+1)/t s=s+t;} return (s); } 若键盘输入n值
为5, 则前n项之和为__2__。
```

参考答案 1: $t=(t+1)/t$ 2: $s=7.76667$

若一自然数等于其所有真因子（能够整除该数，且小于该数的正整数）之和，则称该自然数为完全数。例如，6的真因子有1，2，3，且， $6=1+2+3$ ；因此6为完全数。完成以下代码，实现求[2,1000]之内的完全数个数和最大的完全数，其中，变量n用于存放完全数个数，变量max用于存放最大完全数。#include <iostream.h> int is_complet(int x); //函数is_complet (x)判断x是否为完全数，若“是”，则返回1 void main() { int i,n,max; n=0; for (i=2;i<=1000;i++) if (is_complet(i)==1) { n=n+1; __1__; } cout<<"n="<<n<<endl; cout<<"max="<<max<<endl; } int is_complet(int x) { int n,s=1; for (n=2;n<x;n++) if (x%n==0) __2__; if (x==s) return 1; else return 0; } 运行该程序，最大的完全数是：__3__。

参考答案 1: $\max=i$ 2: $s=s+n$ 3: 496

完成以下程序，实现将十进制数1234转换为八进制数：

```
#include<iostream.h> void main( ) { __1__; f(1234); }
void f(int n) { if (n>=8) f( __2__ ); cout<<n%8; }
```

参考答案 1: void f(int)或void f(int n)或void f(int n);或void f(int); 2: $n/8$ 或 $(n-n\%8)/8$ 或 $(int)(n/8)$

若以下列程序运行时输出结果为：1,A,10.1 2,B,3.5 请将程序补充完整。#include<iostream> using namespace std; int main() { void test(int,char,double__1__); test(1,'A',10.1); test(2,'B'); return 0; } void test(int a, char b,double c) { cout<<a<<','<<b<<','<<c<<endl; }

参考答案 1: $=3.5$

C++语言的参数传递机制包括传值和__1__两种。

参考答案 1: 传地址 或 传址

完成用辗转相除法求最大公约数的函数。

最大公约数辗转相除法算法如下:

- 1) 使得 $m > n$ 。
- 2) m 除以 n 的余数 $\Rightarrow r$
- 3) 如果 $r < 0$, 则 $m=n$; $n=r$ 执行2), 否则执行4)
- 4) n 既最大公约数。

```
int gys(int m, int n)
{
    int r;
    if (m < n) {__1__ } while (r = __2__ )
    {
        m = n;
        n = r;
    }
    return n;
}
```

参考答案

1: $m=m+n$; $n=m-n$; $m=m-n$; 或 $m=m+n$, $n=m-n$, $m=m-n$; 或 $r=m$; $m=n$; $n=r$;
或 $r=m$, $m=n$, $n=r$; 或 $r=n$; $n=m$; $m=r$; 或 $r=n$, $n=m$, $m=r$; 2: $m\%n$ 或 $m-m/n*n$

$f(n)$ 表示从 n 到1 (n 为奇数) 或2 (n 为偶数) 的奇数或偶数的乘积, 完成用递归表示法 $f(n)=n*f(n-2)$ 表示的下列函数。

```
Long f(int n)
{
    long s;
    if ( __1__ ) return 1; __2__
```

```

;
    return s;
}

```

参考答案 1: $n \leq 0$ 2: $s = n * f(n-2)$

函数内定义的动态变量在__1__时才生成。

参考答案 1: 函数调用 或 调用

函数内定义的变量是__1__变量。

参考答案 1: 局部

参数分为形参和__1__。

参考答案 1: 实参

函数名相当于指针常量，它代表函数的__1__。

参考答案

1: 首地址 或 起始地址 或 开始地址 或 入口地址

对于以下递归函数f，调用f(4)的返回值是：__1__。

```

long f(int i)
{
    long r;
    if(i == 0 || i == 1) r = 1;
    else r = i * f(i-1);
    return r;
}

```

参考答案 1: 24

补充完成下面函数，实现求和： $s = 1 + (1+2) + (1+2+3) + \dots + (1+2+3+\dots+n)$ 。

```
long int sum(int n)
{long i, j=0, s=0;
for(i=1; i<=n; i++)
{j=__1__; __2__; return s;}}
```

参考答案 1: j+i 或j+=i 2: s=s+j 或s+=j

如果一个函数中有多个默认参数，则默认参数必须全部处在形参表的__1__部分。**参考答案** 首**或**第一

Vc++6.0中一个工程中可有多个源程序文件，但只能有一个__1__函数。

参考答案 1: main **或** 主

若有函数定义：float f(int x, char y) {…}。将该函数声明为内置函数的语句为__1__。

参考答案 1: inline float f(int, char); **或** inline float f(int, char) **或** inline float f(int x, char y); **或** inline float f(int x, char y)

全局变量定义在__1__位置，包括__2__和静态全局变量两种。其中，后者的作用域不超出定义 它的文件范围，且后者的定义比前者的定义要多一个__3__关键字；而前者的作用域原则上可以扩展到程序所有文件中的所有函数或程序块。

参考答案 1: 函数以外的 2: 外部变量 3: static

static加在局部变量定义前，改变局部变量的__1__但不改变它的__2__。**参考答案** 1: 生存期 2: 作用域

若有宏定义：

```
#define F(a,b) a-b
#define G(a,b) (a+b)
```

程序执行语句中有如下语句：

```
a=6; b=4;
```



```
cout<<F(a,b)*F(a,b)<<G(a,b)*G(a,b)<<G(a,b)/F(a,b)<<endl;
```

则输出的3个数据按次序分别是：

__1__、

__2__、

__3__。

参考答案 1: -22 2: 100 3: -3

在进行函数调用时, 将把参数的__1__传递给值参数, 把参数的__2__传递给引用参数。 **参考答案** 1: 值 2: 地址

假设有如下函数定义: void Fun(int x, int y=100); 则语句Fun(5) 语句__1__等价。 **参考答案** 1: Fun(5, 100)

在函数原型声明中, 必须声明函数参数的类型, 但可以省略__1__。

参考答案 1: 参数名

在C++中, __1__函数不需要使用相应的函数原型语句加以声明。

参考答案 1: main

在调用一个函数的过程中可以直接或间接地调用该函数本身, 这种调用称为__1__调用, 该函数称为__2__。

参考答案 1: 递归 2: 递归函数

C++程序通常由若干个函数组成。C++程序中至少要包含一个__1__。

C++的函数由__2__和 __3__两部分组成。

参考答案 1: main函数 2: 函数首部 3: 函数体

在一个函数中调用另一个函数之前, 一般先要对被调函数提出__1__。

参考答案 1: 声明

从用户使用的角度看, 函数可以分为__1__和__2__。

参考答案 1: 系统函数或库函数或标准函数 2: 用户自定义函数或自定义函数或用户定义函数

在函数的调用过程中，实参与形参间数据的“值传送”方式是指：调用函数时，系统给形参分配__1__，使形参与实参__2__相同的内存空间，形参值的改变不影响实参，实参保持原来的值 不变。

参考答案 1：内存空间 2：不使用

变量的存储类型分为四种：__1__寄存器类型、__2__。

参考答案 1：自动类型或自动存储类型或Auto 2：外部类型或外部存储类型

变量一般需要先赋初值再参与运算，__1__变量没有赋初值时，将由系统自动置为0。

参考答案 1：静态

如果一函数是以return ；结束运行的,说明该函数是__1__函数。参考答案

1：有参

已知函数f的定义是：int f(int n) {n+=1;return 2*n;}

则执行 int m=5,n=3; m=f(n); 后,m 的值是__1__,n的值是__2__。

参考答案 1：8 2：3

、若某函数定义中的函数首部为“int add(int x,int y)”，如果该函数体内return语句后面的表达式值是float类型，在函数的返回值是__1__类型。参考答案 1：int

若要在程序文件中进行标准输入输出操作，则必须在开始的#include命令中使用__1__头文件。

参考答案 1：iostream.h或iostream

本题自定义函数f()的功能是：对给定的参数N, 求 $s=1*1+2*2+\dots+N*N$. 请补充程序中空格处所缺的语句。（注：严格按空格的序号依次书写每空的答案）。

```
#include <iostream>
```

```
using namespace std;
```

```

long f(int N)
{
    long s;
    int i;__1__;
    for(i=1; i<=N; i++)
        s =__2__;
    return s;
}

```

参考答案 1: s=0 2: s+i*i

本题自定义函数f()的功能是：对给定的一个N行N列的二维数组a，求其主对角线上的元素值 之和s. 请补充程序中空格处所缺的语句。

（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>

using namespace std;

#define N 5

long f( int a[][N] )
{
    long s;
    int i, j;__1__;
    for(i=0; i<N; i++)
        for (j=0; j<N; j++)
            { if ( __2__ ) s=s+a[i][j]; }
    return s;
}

```

参考答案 1: s=0 2: i==j

本题自定义函数f()的功能是：对给定的参数N，判断N是否为素数。如果N是素数函数值返回 1，如果N不是素则数函数值返回0。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

long f(int N)
{
    int i, flag;__1__;
    for(i=2; i<=N-1; i++)
        if (__2__==0 ) flag=0;
    return flag;
}
```

参考答案 1: flag=1 2: N % i

本题自定义函数f()的功能是：对给定的参数N，求数列{a[n]}之和s=a[1]+a[2]+...+a[N]；数列 各项的规律是：a[1]=1，其它项满足a[n]=a[n-1]+2. 请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

long f(int N)
{
    long s;
    int n, t;
```

```

    n=1; t=1;__1__;
    while(n<=N)
        { s =s+t;__2__;
n++; }

    return s;

}

```

参考答案 1: s=0 2: t = t+2

如果一函数是以return ; 结束运行的,说明该函数是__1__函数。 参考
答案 1: 有参

已知函数f的定义是: int f(int n) {n+=1;return 2*n;}

则执行 int m=5,n=3; m=f(n); 后,m 的值是__1__,n的值是__2__。

参考答案 1: 8 2: 3

、若某函数定义中的函数首部为“int add(int x,int y)”,如果该函数体内return语句后面的表达式值是float类型,在函数的返回值是__1__类型。

参考答案 1: int

本题自定义函数f()的功能是: 对给定的参数N, 求 $s=1*1+2*2+\dots+N*N$. 请补充程序中空格处所 缺的语句。(注: 严格按空格的顺序号依次书写每空的答案)。#include <iostream>

```

using namespace std;

long f(int N)
{
    long s;
    int i;__1__;
    for(i=1; i<=N; i++)

```

```

        s =__2__;
    return s;
}

```

参考答案 1: s=0 2: s+i*i

本题自定义函数f()的功能是：对给定的一个N行N列的二维数组a，求其主对角线上的元素值 之和s. 请补充程序中空格处所缺的语句。

（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>

using namespace std;

#define N 5

long f( int a[][N] )
{
    long s;

    int i, j;__1__;

    for(i=0; i<N; i++)
        for (j=0; j<N; j++)
            { if ( __2__ ) s=s+a[i][j]; }

    return s;
}

```

参考答案 1: s=0 2: i==j

本题自定义函数f()的功能是：对给定的参数N，判断N是否为素数。如果N是素数函数值返回 1，如果N不是素则数函数值返回0。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>

```

```

using namespace std;

long f(int N)
{
    int i, flag;__1__;
    for(i=2; i<=N-1; i++)
        if (__2__==0 ) flag=0;
    return flag;
}

```

参考答案 1: flag=1 2: N % i

本题自定义函数f()的功能是：对给定的参数N，求数列{a[n]}之和s=a[1]+a[2]+...+a[N]；数列 各项的规律是：a[1]=1，其它项满足a[n]=a[n-1]+2. 请补充程序中空格处所缺的语句。（注：严 格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>

using namespace std;

long f(int N)
{
    long s;
    int n,t;
    n=1; t=1;__1__;
    while(n<=N)
        { s =s+t;__2__;
n++; }
    return s;
}

```

```
}
```

参考答案 1: s=0 2: t = t+2

本题自定义函数f()的功能是：设 $N1 < N2$ 为正整数，要求统计区间 $[N1, N2]$ 中有多少个整数符合条件“能被5整除但不能被3整除”；并输出符合条件的数，函数的返回值是符合条件的数据个数。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

long f(int N1, int N2)
{__1__;
    int i;
    for (i=N1; i<=N2; i++)
        if (i%5==0__2__i%3!=0)    n++;
    return n;
}
```

参考答案 1: long n=0 2: &&

本题自定义函数f()的功能是：用递归函数求a的n次幂（设a是非零数，n是非负整数 $n \geq 0$ ）。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

double f(float a, int n)
{__1__;
    if (n==0) y = 1;
```



```

        else __2__;
        return y;
}

```

参考答案 1: double y 2: y=a*f(a,n-1)

本题自定义函数f()的功能是：用递归函数求n的阶乘n!（设n是非负整数 $n \geq 0$ ， $0! = 1$ ）。请补充 程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>
using namespace std;
double f(int n)
{ __1__;
    if (n==0) y = 1;
    else __2__;
    return y;
}

```

参考答案 1: double y 2: y=n*f(n-1)

本题自定义函数f()的功能是：用递归函数求[1, n]区间中所有整数之和（ $n \geq 1$ ）。请补充程序中 空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>
using namespace std;
double f(int n)
{__1__;
    if (n==1) y = 1;

```

```

        else__2__;
    return y;
}

```

参考答案 1: double y 2: y=n+f(n-1)

本题自定义函数f()的功能是：对给定的一维数组a及其元素个数n两个参数，找出数组中的最大元素及其下标，最大值由全局变量maxnum存储，对应元素下标由函数值返回。如果a中有多 个并列最大的元素，以第一次出现的元素为准。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>

using namespace std;__1__;

int f(long a[], int n)
{
    int i, maxI;
    maxnum=a[0];
    for(i=1; i<n; i++)
        if ( __2__ )
        {maxnum = a[i]; maxI = i ;}
    return maxI;
}

```

参考答案 1: long maxnum 2: a[i]>maxnum

本题自定义函数f()的功能是：对给定的一维数组a及其元素个数n两个参数，找出数组中的最小元素及其下标，最小值由全局变量minnum存储，对应元素下标由函数值返回。如果a中有多 个并列最小的元素，以第一次出现的元素为准。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案，相邻两空的答案之间用英文逗号

隔开，不要加额外空 格）。

```
#include <iostream>

using namespace std; __1__;

int f(long a[], int n)
{
    int i, minI;
    minnum=a[0];
    for(i=1; i<n; i++)
        if (__2__)
            {minnum = a[i]; minI = i ;}
    return minI;
}
```

参考答案 1: long minnum 2: a[i]<minnum

本题自定义函数f()的功能是：对给定的一维数组a及某个指定数x，找出数组a中元素值等于x 的元素个数，以及值为x且下标最小的那个元素的下标。下标最小的那个元素的下标在函数中 输出；而元素值等于x的元素个数作为函数值返回。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

int f( long a[], int n, long x) //n是数组a的元素个数
{
    int i, minI, s;
    s=0;
```

```

        minI=-1;
    for(i=1; i<n; i++)
        if ( __1__ )
            s = s+1;
        for(i=1; i<n; i++)
            if ( a[i]==x && minI<0 ) __2__ ;
    cout<<"minI="<<minI<<endl;
    return s;
}

```

参考答案 1: a[i]==x 2: minI= i

本题自定义函数f()的功能是：对给定的一维数组a及其元素个数n，求偶数下标的元素之和以及奇数下标的元素之积。偶数下标的元素之和在函数中输出；而奇数下标的元素之积作为函数值返回。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>
using namespace std;
int f( long a[], int n) //n是数组a的元素个数
{
    int i, s1, s2;
    s1=0; s2=1;
    for(i=1; i<n; i++)
        if ( __1__ ==0)
            s1 = s1+a[i];
        else

```

```

        s2 =__2__;
        cout<<"s1="<<s1<<endl;

        return s2;
    }

```

参考答案 1: i%2 2: s2*a[i]

本题自定义函数f()的功能是用辗转相除法求m、n两数的最大公约数，函数值返回它们的最大 公约数。请补充程序中空格处所缺的语句。
(注：严格按空格的顺序号依次书写每空的答案)。

```

#include <iostream>

using namespace std;

int f(int m, int n)
{ int t, r;
    if (m<n)
        { t=m; m=n; n=t;}__1__;
    while(r!=0)
        { m=n;__2__;
            r=m%n; }
    return n;
}

```

参考答案 1: r=m%n 2: n=r

本题自定义函数f()的功能是：给定数组x及其元素个数n，将数组x的元素反序存放。反序的操作是将第1个位置的元素与倒数第1个位置的元素交换，将第2个位置的元素与倒数第2个位置的 元素交换，…按这个规律做下去，交换的元素组数与元素总数n有关。请补充程序中空格处所 缺的语句。(注：严格按空格的顺序号依次书写每空的答案)。

```

#include <iostream>
using namespace std;
void f(int x[], int n)
{ int i, j, t;
for (i=0, j=n-1; i<=__1__; i++, j--) {t=x[i]; __2__;
x[j]=t;}
}

```

参考答案 1: $n/2-1$ 2: $x[i]=x[j]$

本题自定义函数f()的功能是：求满足下列二元一次方程组的正整数解并输出所有的正整数解。 $x+y=M$, $1.5x+5.5y=N$ 。其中，M和N是将给定的两个确定参数。（注：严格按空格的顺序 号依次书写每空的答案）。

```

#include <iostream>
using namespace std;
void f(int M, int N)
{int x, y;
for (x=1; x<= __1__; x++)
{ y=M-x;
if (1.5*x+5.5*y __2__ N)
cout<<x<<', '<<y<<endl;
}
}
}

```

参考答案 1: $M-1$ 2: $==$

本题自定义函数f()的功能是：对给定的三位正整数N，使得函数返回

值是原数的反序数，如 $N=123$ ，则返回321；如果主调语句给定的实参不是三位的正整数，则函数返回值将为-1。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

int f(int N)
{ int x, y, z;

  int M;

  if (N<100__1__999 ) return -1;

  x=N%10;

  y=__2__;          z=__3__;

  M=100*x+10*y+z;

  return M;

}
```

参考答案 1: $||N>$ 2: $(N/10)\%10$ 3: $N/100$

本题自定义函数f()的功能是：对给定的三位正整数N，使得函数返回值是原数的反序数，如 $N=123$ ，则返回321；如果主调语句给定的实参不是三位的正整数，则函数返回值将为-1。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

int f(int N)
{ int x, y, z;

  int M;

  if (N<100__1__999 ) return -1;
```

```

    x=N%10;

    y=__2__;        z=__3__;

    M=100*x+10*y+z;

    return M;

}

```

参考答案 1: `||N>` 2: `(N/10)%10` 3: `N/100`

本题自定义函数f()的功能是：对给定的三位正整数N，判断N是否为“水仙花数”，水仙花数是三位的正整数，且各数位上数字的立方和等于该数本身。如果主调语句给定的实参不是三位的正整数，则函数返回值将为-1；如果主调语句给定的实参是三位的正整数但不是水仙花数，则函数返回值将为0；如果主调语句给定的实参是水仙花数，则函数返回值将为1。请补充程序中空格处所缺的语句。（注：严格按照空格的顺序号依次书写每空的答案）。

```

#include <iostream>

using namespace std;

int f(int N)
{
    int x, y, z;

    int M;

    if (N<100 __1__ 999 ) return -1;        x=N%10; y=__2__;
    z=N/100;        if (x*x*x+y*y*y+z*z*z__3__) return 1;

    else return 0;

}

```

参考答案 1: `||N>` 2: `(N/10)%10` 3: `==N`

本题自定义函数f()的功能是：对给定的n位正整数N（假定数的位数n不超过5位），判断N是否为“回文数”，回文数是指正读和反读都一样的正整数，如34143就是一个4位的回文数。如果N是回文数，则函数返回值为1；否则，函数返回值为0。请补充程序中空格处所缺的语

句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

int f(int n, int N)
{ int i,M,a[5];

  M=N;

  for (i=0;i<n;i++) //此循环求出N的各位数字存放在数组a中
  { a[i]=M%10;__1__;
  }

  //以下循环求出N的反序数，结果存放在M中

  M=0;

  for (i=0;i<n;i++)

    M=M*10+__2__;    if (M__3__) return 1;

  else return 0;

}
```

参考答案 1: M=M/10 2: a[i] 3: ==N

本题自定义函数f()的功能是：对一给定的正整数N，求出其各数位上的数字之和作为函数返回值，如N=135时，函数返回值为9。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

long f(int N)

{ int i,r;
```

```

long s=0;
while(N__1__)    { __2__;
    s=s+r;
    N= __3__;
}

return s;
}

```

参考答案 1: !=0 2: r=N%10 3: N/10

本题自定义函数f()的功能是：对一给定的正整数N，求出其各数位上的数字之积作为函数返回值，如N=235时，函数返回值为30。请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```

#include <iostream>

using namespace std;

double f(int N)
{ int i,r;__1__;
  while(N!=0)
  { __2__;
    s=s*r;
    N=__3__;
  }

  return s;
}

```

参考答案 1: double s=1 2: r=N%10 3: N/10

本题自定义函数f()的功能是：对一给定的正整数N，求出其各数位上的最大数字，如N=2385 时，函数返回值为8。 请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写 每空的答案）。

```
#include <iostream>

using namespace std;

int f(long N)
{ int i,r,max=0;
  while(N!=0)
  { __1__;          if (r>max) __2__;__3__;
  }

  return max;
}
```

参考答案 1: r=N%10 2: max=r 3: N=N/10

本题自定义函数f()的功能是：对一给定的正整数N，求出该数的组成数字中有多少个是偶数数字，结果作为函数值返回。如N=23826时，函数返回值为4。 请补充程序中空格处所缺的语句。（注：严格按空格的顺序号依次书写每空的答案）。

```
#include <iostream>

using namespace std;

int f(long N)
{ int i,r,s=0;
  while(N!=0)
  { r= __1__;          if (r__2__) s++;          N=__3__;
  }

  return s;
}
```

```
}
```

参考答案 1: N%10 2: %2==0 3: N/10

本题自定义函数f()的功能是：对给定的正整数N (N<10)，生成一个N×N的规则方阵的下三角部分的数据（含主对角线上元素），其它元素取默认0值；然后输出整个方阵的元素。如当 N=5时，输出方阵如本题图所示。（注：严格按空格的顺序号依次书写每空的答案）。

00000

12000

23400

34560

45678

```
#include <iostream>
```

```
using namespace std;
```

```
void f(long N)
```

```
{ int i, j;
```

```
static a[9][9]; //静态数组不初始化，各元素的默认初值均为0
```

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

```
for(j=0; j<=__1__; j++)                    a[i][j]=i+__2__;
```

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

```
{for(j=0; j<__3__; j++)
```

```
cout<<a[i][j]<<' ';
```

```
cout<<endl;}
```

```
return ;
```

```
}
```

参考答案 1: i 2: j 3: N

本题自定义函数f()的功能是：对给定的正整数N (N<10)，生成一个N×N的规则方阵。如当N =5时，方阵如本题图所示。（注：严格按空格的顺序号依次书写每空的答案）。

12345

11234

12123

13312

14641

```
#include <iostream>
```

```
using namespace std;
```

```
void f(long N)
```

```
{ int i, j, a[9][9];
```

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

```
        for(j=0; j<N; j++)
```

```
            { if (i__1__0) a[i][j]=1;          else if (i<j) a[i][j]=  
1+__2__;          else a[i][j]=a[i-1][j]+__3__;
```

```
    }
```

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

```
        {for(j=0; j<N; j++)
```

```
            cout<<a[i][j]<<' ';
```

```
            cout<<endl;}
```

```
    return ;
```

```
}
```

参考答案 1: `==j||j==` 2: `j-i` 3: `a[i-1][j-1]`

本题自定义函数f()的功能是：对给定的正整数N (N<10)，生成一个N行的字符图案。如当N =5时，图案如本题图所示。（注：严格按空格的顺序号依次书写每空的答案）。

K

okkk

ookkkkk

ookkkkkkk

oooookkkkkkkkk

```
#include <iostream>
```

```
using namespace std;
```

```
void f(long N)
```

```
{ int i, j;
```

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

```
        {for(j=0;j<__1__;j++)
```

```
            cout<<'o' ;
```

```
            for(j=0;j<__2__;j++) __3__;
```

```
cout<<endl;
```

```
    }
```

```
    return ;
```

```
}
```

参考答案 1: `i` 2: `2*i+1` 3: `cout<<'k'`

若有以下定义：

```
static int x;
```

则变量x的初始值是

__1__。

参考答案 1: 0

以下程序的功能是：在主函数中指定t值，通过调用自定义函数求Fibonacci数列中大于指定数t 的最小一个数。其中Fibonacci数列F(n)的定义为：F(0)=0, F(1)=1, ……，F(n)=F(n-1)+F(n-2)

例如，当t=1000时，函数值为1597。

将程序补充完整。

```
include <iostream>

using namespace std;

unsigned int fibo(int);

int main() {
    cout<<"input a number:";
    unsigned int t,i=0,m;
    cin>>t;
    while((m=
__1__)<t)
        i++;
    cout<<m<<endl;
    return 0;
}

unsigned int fibo(int n) {
    unsigned int y;
    if(n==0)
```

```

y=0;
else if(n==1)
    y=1;
else
    __2__ return y;}

```

参考答案 1: fibo(i) 2: y=fibo(n-1)+fibo(n-2);

求 $s=3+33+333+\dots+333\dots3$ 的和，即第 n 个加数是一个 n 位的整数，且各数位上的数字均为3。要求设计一个递归函数实现求每一个加数，然后用累加求和方式求出总和。将程序补充完整。

```

#include <iostream>

using namespace std;

unsigned int  number(int);

int main() {
    unsigned int s=0;
    int m;
    cout<<"input m:";
    cin>>m;
    for(int i=1;i<=m;i++)
        s+=
        __1__;
    cout<<"s="<<s<<endl;
    return 0;
}

unsigned int  number(int n) {

```



```

unsigned int y;
if(n==1)
    y=3;
else
    y=
__2__;
return y;
}

```

参考答案 1: number(i) 2: 10*number(n-1)+3

下列程序的功能是求[1000, 2000]之间最小的素数和最大的素数。将程序补充完整。

```

#include <iostream>
#include <cmath>
using namespace std;
bool is_prime(unsigned int);
int main() {
    unsigned int min(2000), max(1000);
    for(unsigned int i=1000; i<=2000; i++)
        if(is_prime(i)) {
            if(i<min)
                __1__ if(i>max) __2__
        }
    cout<<"min="<<min<<endl;
    cout<<"max="<<max<<endl;
}

```

```

return 0;
}

bool is_prime(unsigned int x)
{
    bool z=true;
    unsigned int m=int(sqrt(x));
    for(unsigned int n=2;
        __3__;n++)
        if(!(x%n))
        {
            z=false;
            break;
        }
    return z;
}

```

参考答案 1: min=i; 2: max=i; 3: n<=m

回文数是指正读和反读都一样的正整数。例如，5，121等都是回文数。求[1, 500]以内的回文数 数目及所有回文数之和。将程序补充完整。

```

#include <iostream>

using namespace std;

bool palindrome(unsigned int);

int main() {
    unsigned int k=0,s=0;

```

```

for(unsigned int m=1;m<=500;m++)
if(palindrome(m))
k++, s+=m;
cout<<"1-500以内的回文数有"<<k<<"个，所有回文数之和"<<s<<endl;
return 0;
}

bool palindrome(unsigned int n){
unsigned int i=n,m=0;
while( i ) {
m=
__1__i/=10;
}
__2__}

```

参考答案 1: $m*10+i\%10$;或 $10*m+i\%10$;或 $i\%10+m*10$;或 $i\%10+10*m$; 2:
return (m==n);或return (n==m);或return n==m;或return m==n;

一个变量的引用变量名是该变量的__1__。 参考答案 1: 别名

函数参数有两种传递方式：传值和__1__。

参考答案 1: 传引用或传地址或传址

函数名相当于指针常量，它代表函数的__1__。

参考答案 1: 起始地址 或 开始地址

若有函数定义：

```

bool isNum(char ch)
{ return (ch>= '0' && ch<= '9' ) ; }

```

将该函数声明为内联函数的语句为：__1__

参考答案 1: inline bool isNum(char ch);或inline bool isNum(char);

以下程序段是函数返回 a 所指数组中最小的值所在的下标值，请在下划线处填空。

```
int fun(int a[], int n) {  
    int i, p=0;  
    for(i=1; i<n; i++)  
        if(a[i]<a[p])  
            __1__;  
    return p;  
}
```

参考答案 1: p=i;

函数的定义或声明前加上关键字__1__，就会把该函数定义为内联函数

参考答案 1: inline

若已定义“int a[10]”，以下fun函数的功能是：在第一个循环中给前 10 个数组元素依次赋1、 2、 3、 4、 5、 6、 7、 8、 9、 10；在第二个循环中使 a 数组前10个元素中的值对称折叠，变成 1、 2、 3、 4、 5、 5、 4、 3、 2、 1。请填空。

```
void fun(int a[ ]){  
    int i;  
    for(i=1; i<10; i++)  
        __1__=i;  
    for(i=0; i<5; i++)  
        __2__=a[i];  
}
```

```
}
```

参考答案 1: $a[i-1]$ 2: $a[9-i]$

以下程序输出的最后一个值是__1__。

```
#include<iostream>
using namespace std;
int ff(int n){
    static int f=1;
    f*=n;
    return f;
}
main(){
    int i;
    for(i=1;i<=5;i++)
        cout<<ff(i);
}
```

参考答案 1: 120

以下函数的功能是求x的y次方，请填空。

```
double fun(double x, int y)
{
    int i;
    double z;
    for(i=1, z=x; i<y; i++)
        __1__;
    return z;
}
```

```
}
```

参考答案 1: $z=z*x$

采用函数重载的目的是__1__。 参考答案 1: 使用方便, 提高可读性
具有相同函数名但具有不同参数表函数称为__1__。

参考答案 1: 重载函数

实现比较两个字符串大小的函数是__1__。 参考答案 1: strcmp

在调用f1()函数过程中又调用了f1()函数, 这种调用称为__1__。 参考
答案 1: 直接递归; 直接递归调用

内联函数是通过__1__来实现的。 参考答案 1: 编译器

程序总是从__1__函数开始执行。 参考答案 1: 主或main或main()

函数定义中, 出现在函数首部的参数是__1__参数 参考答案 1: 形式

函数调用语句中的参数叫__1__参数 参考答案 1: 实在或实际

阅读以下程序:

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

```
int a=5,b=3;
```

```
max(inta ,intb)
```

```
{
```

```
    int c;
```

```
    c=a>b?a:b;
```

```
    return(c);
```

```
}
```

```
void main()
```

```
{
```

```

    int a=8;
    cout<<" max=" <<max(a, b)<<endl:
}

```

写出其运行结果:

参考答案 max=8

阅读以下程序:

```

#include    <iostream.h>

int  x;

void  fun()
{
    Cout<<x++<<endl;
}

void  main()
{
    int(x=1);
    if(x==1)
    {
        int x=2;
        cout<<x++<<'    ' ;
    }
    fun();
}

```

写出其运行结果:

参考答案 2 0

阅读以下程序：

```
#include <iostream.h>

int kk()
{
    int x=4;
    static int y;
    y=5;
    x*=2;
    u*=2;
    return(x+y);
}

void main()
{
    int j, s=0;
    for(j=0; j<2; j++)
        s=kk();
    cout<<" s=" <<s<<endl;
}
```

写出当j=1时，程序运行时的输出结果：

参考答案 s=18

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



```

int main()
{int i;
for(i=0;i<4;i++)
cout<< f(i) <<' \t' ;
return 0;}

int f(int i)
{static int k=1;
for(;i>0;i--)
k+=i;
return k;}

```

参考答案 1 2 5 11

```

#include <iostream>
using namespace std;
int max(int x,int y)
{ float m;
m=x>y?x:y;
x=2*x;y=y+1;
cout<<x<<" "<<y<<endl;
return (m);
}

void main()
{float a,b,c;
a=3.0;b=8.0;

```

```
c=max(a,b);  
cout <<c<<" "<<a<<" "<<b<<endl;  
}
```

参考答案 6,9 8,3,8

```
#include <iostream.h>  
  
int func(unsigned char x)  
{  
    int countx = 0;  
    while(x)  
    {  
        countx ++;  
        x = x&(x-1);  
    }  
    return countx;  
}  
  
void main()  
{  
    cout<<func(255)<<endl;  
}
```

参考答案 8

```
#include <iostream.h>  
  
#define F(X,Y) (X)*(Y)  
  
void main()
```

```
{ int a=3, b=4;
cout<< F(a++, b++)<<endl;
}
```

参考答案 12

```
#include <iostream.h>
fun(int x)
{ if(x/2>0) fun(x/2);
cout<<x<<endl;
}
main()
{ fun(6); }
```

参考答案 1 3 6

```
#include <iostream.h>
fun(int a, int b)
{ if(a>b) return(a);
else return(b);
}
main()
{ int x=3, y=8, z=6, r;
r=fun(fun(x, y), 2*z);
cout<< r<<endl;
}
```

参考答案 12

```

#include <iostream.h>

int a=5;

void fun(int b)
{
    static int a=10;
    a+=b++;
    cout<<a;
}

void main( )
{
    int c=20;
    fun(c);
    a+=c++;
    cout<<a<<endl;
}

```

参考答案 3025

```

#include <iostream.h>

int f(int n)
{if(n==1)return 1;
else return f(n-1)+1;
}

main()
{int i, j=0;

```

```
for(i=1;i<5;i++) j+=f(i);  
cout<<j;  
}
```

参考答案 10

```
#include <iostream.h>  
  
int func(int a)  
  
{ int b=0; static c=3;  
  
b++;c+=1;  
  
return(a+b+c);  
  
}  
  
main( )  
  
{ int a=4, i;  
  
for(i=0;i<3;i++)  
  
cout<<func(a);
```

```
}
```

参考答案 91011

阅读程序，写出运行结果。

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

```
int d=1;
```

```
void fun(int p)
```

```
{
```

```
    int d=5;
```

```
    d+=p++;
```

```
    cout<<d;
```

```
}
```

```
void main( )
```

```
{
```

```
    int a=3;
```

```
    fun(a);
```

```
    d+=a++;
```

```
    cout<<d;
```

```
}
```

参考答案 84

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

```
int foo(void)
```

```
{
```

```
    unsigned int a = 6;
```

```

        int b = -20;
        b=(a+b>6)?0:1;
        return b;
    }

```

```

main()
{
    cout<<foo();
}

```

参考答案 0

```

#include <iostream.h>
main()
{ int i,j,k; int func(int x,int y);
i=1, j=2, k=3;
cout<<func((j+=k, j++), (i, i++, ++i));
}
int func(int x,int y)
{int z;
z=x%y;
return(z); }

```

参考答案 2

运行下面程序，从键盘输入四个字符xyz#，其输出是 。

```

#include <stdio.h>
main()

```

```

{
void  recursion();
recursion();
}

void  recursion()
{
char c;
c=getchar();
putchar(c);
if(c!=' #' )
recursion();
putchar(c);
}

```

参考答案 xyz##zyx

```

#include <iostream.h>

#define EVEN(x)  x%2==0?1:0

main( )

{   if(EVEN(9+1)) cout<<0;

    else cout<<1;    }

```

参考答案 1

```

#include <iostream.h>

int f(int a)

{static int c=0;

```



```

c=a+c++;
return(c);
}

void main() {int a=2, i, k;
for(i=0; i<2; i++)
k=f(a++);
cout<<k;}

```

参考答案 7

```

#include <iostream.h>

int m=13;

int fun(int x, int y)
{int m=3;
return(x*y-m);
}

main()
{int a=7, b=5;
cout<<fun(a, b)/m;
}

```

参考答案 2

```

#include <iostream.h>

int f1(int x, int y) {return x>y?x:y;}
int f2(int x, int y) {return x>y?y:x;}

main()

```

```

{
int a=4, b=3, c=5, d=2, e, f, g;
e=f2(f1(a, b), f1(c, d)); f=f1(f2(a, b), f2(c, d));
g=a+b+c+d-e-f;
cout<<e<<f<<g;
}

```

参考答案 437

```

#include <iostream.h>

#define P 3

void F(int x){return(P*x*x);}

main() {cout<<F(3+5);}

```

参考答案 192

```

#include <iostream.h>

#define S(x) 4*x*x+1

main()
{
int i=6, j=8;
cout<<S(i+j);
}

```

参考答案 81

```

#include <iostream.h>

int fun(int a)
{int b=0;static int c=3;

```

```

b++;c++;return(a+b+c);
}

main()
{int i,a=5;
for(i=0;i<3;i++)cout<<i<<"  "<<fun(a)<<" ";

}

```

参考答案 0 10 1 11 2 12

```

#include <iostream.h>

int Xfun(int x){
int y=x;
{int x=10; y+=x;}
return x+y;
}

void main()
{cout<<Xfun(5);
}

```

参考答案 20

```

#include<iostream>

using namespace std;

void foo(int a, int b, int c)
{
    if(a > b)

```

```

    foo(c, a, b);
else
    cout<<a<<','<<b<<','<<c<<endl;
}

int main()
{
    int a=2, b=1, c=3;
    foo(a, b, c);
    return 0;
}

```

参考答案 1, 3, 2

```

#include<iostream>
using namespace std;
void f(int a, int b, int c)
{
    cout << a << b << c << '/' ;
    if(a == 3 && b == 2 && c == 1)
        return;
    if(b < c)
        f(a, c, b);
    else if(a < b)
    {
        if(a < c)

```

```
        f(c, a, b);  
    else  
        f(b, c, a);  
    }  
}
```

```
int main()  
{  
    int a=1, b=3, c=2;  
    f(a, b, c);  
    cout << endl;  
    return 0;  
}
```

参考答案 132/213/231/312/321/

```
#include<iostream>  
using namespace std;  
float FH(float x) {  
  
    float y=0, n=0;
```

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

```
n++; y+=x;
```

```
x=(int)x;
```

```
}
```

```
return y/n;
```

```
}
```

```
int main()
```

```
{
```

```
cout << FH(3.12)<<endl;
```

```
return 0;
```

```
}
```

参考答案 3.024

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

```
#define ADD(x) (x)+(x)
```

```
main()
```

```
{    int  a=4, b=6, c=7, d=ADD(a+b)*c;
```

```
    cout<<d;
```

```
}
```

参考答案 80

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

```
int hcf(int u,int v)
```

```
{ int  a,b,c,r;
```

```
    if(u>v)    { c=u; u=v; v=c; }
```

```
    a=u; b=v;
```

```
    while((r=b%a)!=0) { b=a; a=r;}
```

```
    return(a);
```

```
}
```

```
main()
```

```
{ int u, v, h, l;
```

```
u=32; v=48;
```

```
h=hcf(u, v);
```

```
cout<<h;
```

```
}
```

参考答案 16

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

```
main()
```

```
{int fact(int value);
```

```
cout<<fact(5)<<endl;
```

```
cout<<fact(-5)<<endl;
```

```
}
```

```
int fact(int value)
```

```
{ int f;
```



```

    if(value<0)
    {cout<<"Arg error\n";
        return(-1);

    }

    else if(value==1||value==0)    f=1;

    else f=value*fact(value-1)+1;

    cout<<"This called value="<<f<<"\n";

    return f;

}

```

参考答案 This called value=1 This called value=3 This called value=10 This called value=41 This called value=206 206 Arg error -1

struct sd{ int a; double b;}s;语句定义了变量s,s存储单元的长度至少为_____。

A 6 B 12 C 16 D 14 **参考答案** B

将圆的周长计算式定义为一个宏L, 正确的写法为_____。

A #include 2*PI*R L B #include L 2*PI*R C #define 2*PI*R L(R) D #define L(R) (2*PI*(R)) **参考答案** D

在面向对象的程序设计C++中, 当用cin、cout输入、输出时程序头部

要_____。

A #include<stdio.h> B #include<iostream.h>

C #include<stdlib.h> D #include<string.h> 参考答案 B

源程序中包含文件编译命令的作用是_____。

A 定义格式 B 条件编译 C 把某个头文件中已写好的源代码加入源程序

D 保护结构 参考答案 C

以下程序段的输出结果是（ ）。

```
#include<iostream>
using namespace std;
#define min(x,y) (x)<(y)?(x):(y)
void main() {
    int i=10, j=15, k;
    k=10*min(i, j);
    cout<<k<<endl;
}
```

A 15 B 100 C 10 D 150 参考答案 A

以下叙述中不正确的是（ ）。

A C++程序中的预处理命令行都以#号开始 B 宏定义中的宏名可用大写字母或小写字母 C 宏替换不占用运行时间，只占编译时间 D 使用带参数的宏时，参数的类型应与宏定义时的一致 参考答案 D

以下正确的描述为

A 每个C++语言程序必须在开头用预处理命令 B 预处理命令必须位于C++源程序的首部 C 在C++语言中预处理命令都以”#”开头 D C++语言的预处理命令只能实现宏定义和条件编译的功能 参考答案 C

利用表达式a[i]可以访问int型数组a中下标为i的元素。在执行了语句int *p=a;后，利用指针p也可访问该元素，相应的表达式是__1__。参

参考答案 1: `*(p+i)`

程序中的预处理命令是指以__1__字符开头的命令。

参考答案 1: `#`

对源程序进行编译时将进行__1__分析、语法分析和逻辑规则分析。参

参考答案 1: 词法

预处理语句有3种，分别是宏定义、__1__和条件编译。

参考答案 1: 文件包含

已知数组arr的定义如下：`int arr[5]={1, 2, 3, 4, 5}`；下列语句中输出结果不是2的是

A `cout << *arr+1 << endl`; B `cout << * (arr+1) << endl`; C `cout << arr[1] << endl`; D `cout << *art << endl`; 参考答案 D

已知类MyClass 声明如下：

```
class MyClass {  
    int n;  
public:  
    MyClass(int k) : n(k) {}  
    int getValue()const { return n; }  
};
```

在下列数组定义中正确的是

A `MyClass x1[2]`; B `MyClass x2[2]={new MyClass (1) , new MyClass (2) }`; C `MyClass *x3[2]`; D `MyClass *x4[2]={MyClass (1) , MyClass (2) }`; 参考答案 C

已知有数组定义`char a[3][4]`；下列表达式中错误的是 A `a[2]="WIN"` B `strcpy(a[2], "WIN")` C `a[2][3]='W'` D `a[0][1]=a[0][1]` 参考答案 C

要定义一个具有5个元素的一维数组vect，并使其各元素的初值依次是

30、50、-88、0和0，则下列定义语句中错误的是（ ）。

A `int vect[]={30,50,-88};` B `int vect[]={30,50,-88,0,0};`

C `int vect[5]={30,50,-88};` D `int vect[5]={30,50,-88,0,0};` 参考答案 A

有下列程序：`#include<iostream.h> void main() { int i,k,a[10],p[3]; k=5; for(i=0;i<9;i++) a[i]=i; for(i=0;i<3;i++) p[i]=a[i*(i+1)]; for(i=0;i<3;i++) k+=p[i]*2; cout<<k; }` 其输出结果是（ ）。

A 20 B 21 C 22 D 23 参考答案 B

设有数组定义：`char array[]="China";`，则数组array所占的空间为_____。

A 4个字节 B 5个字节 C 6个字节 D 7个字节 参考答案 C

在`int a[][3]={ {1}, {3,2}, {4,5,6}, {0} }`中，`a[2][2]`的值是_____。

A 3 B 2 C 6 D 4 参考答案 C

下面关于一维数组的说法错误的是（ ）。

A 一维数组的定义中，表示数组长度的可以是直接常量、符号常量和变量。 B 数组元素的下标从0开始。 C 对全部数组元素都赋初值时，可以不指定数组长度。 D 数组代表具有同种类型的一组数据。 参考答案 A

对二维数组的正确定义是（ ）。

A `int a[2,3]={1,2,3,4,5,6};` B `int a[2][]={1,2,3,4,5,6};` C `int a[][]={1,2,3,4,5,6};` D `int a[][3]={1,2,3,4,5,6};`

参考答案 D

下列数组定义中，不正确的是（ ）。

A `int x[1][3];` B `int x[2][2]={1,2,3};` C `int x[2][]={1,2,3,4};` D `int x[][2]={1,2,3,4}` 参考答案 C

如有定义：int a[3][4]；则下面可以正确引用数组元素的表达式是（ ）。

A a[2][4] B a[1,3] C a[2][0] D a(2)(1) 参考答案 C

如有定义：int a[20]；则下面可以正确引用数组元素的表达式是（ ）。

A a[20] B a[3.5] C a(5) D a[10-10] 参考答案 D

设有int x[][3]={ {0}, {1,2}, {3,4,6}, {5}}； 则x[2][2]的值是（ ）。

A 3 B 2 C 6 D 4 参考答案 C

在下面的二维数组定义中，正确的是（ ）。

A int a[3][]; B int a[][3]; C int a[][3]={ {1,3,5}, {2}}; D int a(3)(3); 参考答案 C

下面关于二维数组的说法错误的是（ ）。 A 同行的数据在内存中连续存放。 B 对全部数组元素都赋初值时，可以不指定数组第二维的长度。 C 不能将二维数组作为一个整体输入与输出。 D 将二维数组看作是一维数组时，该一维数组的每个元素本身又是一个一维数组。 参考答案 B

已知char a[][20]={ "beijing", "shanghai", "changsha" };语句cout<<a[3]; 得到的输出是：

A j B changsha C 输出结果不确定 D 数组定义有错 参考答案 C

要定义一个具有5个元素的一维数组vect，并使其各元素的初值依次是30，50，-88，0和0，则下列定义语句中错误的是（ ）。

A int vect[]={30,50,-88}; B int vect[]={30,50,-88,0,0}; C int vect[5]={30,50,-88}; D int vect[5]={30,50,-88,0,0}; 参考答案 A

若有定义 int a[10]; ，则引用a[-1]将（ ）。

A 导致编译错误 B 导致运行错误 C 产生不确定结果 D 可编译通过，但不能运行 参考答案 C

C++规定： 对于一个具有N个元素的一维数组，其下标的取值范围是（ ）。

A $0 \sim N$ B $1 \sim N$ C $0 \sim N-1$ D $N \sim 1$ 参考答案 C

引用一个数组元素时，数组元素的下标表达式的类型（ ）。

A 必须是整型常量 B 必须是整型表达式 C 必须是整型常量或整型表达式 D 可以是任何类型的表达式 参考答案 D

若函数形参是数组，则对应的实参（ ）。

A 只能是数组名 B 只能是指针 C 可以是数组名或指针 D 任何类型的数据 参考答案 C

在下列对字符数组进行初始化中（ ）是正确的。

A `char s1[]="abcd"` ; B `char s2[3]="xyz"` ; C `char s3[][]={{'a','x','y'}}` ; D `char s4[2][3]={ "xyz", "mnp" }` ; 参考答案 A

以下能正确定义一维数组的是()

A `int num[]` ; B `#define N 100` C `int num[0...100]` ; D `int N=100`
`int num[N]` ; 参考答案 D

有如下数组声明“`int values[30];`”，下标值引用错误的是()

A `values[30]` B `values[20]` C `values[10]` D `values[0]` 参考答案 A

以下选项中不能正确赋值的是()

A `char s1[10];s1="cbest"` ; B `char s2[]={ 'c', 'b', 'e', 's', 't' }` ; C `char s3[10] ="cbest"` ;
D `char s4 ="cbest\n"` ; 参考答案 A

对于“`int *pa[5];`”的描述,()是正确的。

A `pa`是一个指向数组的指针，所指向的数组是5个int型元素。 B `pa`是一个指向某数组中第5个元素的指针，该元素是int型。 C `pa[5]`表示某个数组的第5个元素的值。 D `pa`是一个具有5个元素的指针数组，每个元素是一个int型指针。 参考答案 D

在 “int b[4][4]={ {1}, {2, 3}, {4, 5, 6}, {7, 8, 9, 10}}; ” 中b[3][3]的值是 ()。

A 0 B 10 C 6 D 9 参考答案 B

以下各组选项中，均能正确定义二维实型数组 a 的选项是 ()。

A float a[3][4]; float a[][4]; float a[3][]={{1}, {0}};

B float a(3,4); float a[3][4]; float a[][]={{0}, {0}};

C float a[3][4]; float a[][4]={{0}, {0}}; float a[][4]={{0}, {0}, {0}};

D float a[3][4]; float a[3][]; float a[][4]; 参考答案 C

若有语句 “int a[3][4]={0};”，则下面正确的叙述是 ()。 A 只有元素a[0][0]可得到初值0 B 数组a中各元素都可得到初值，但其值不一定为0 C 数组a中每个元素均可得到初值0 D 此语句不正确。 参考答案 C

若有以下程序：

```
void main() {  
    int a[]={4, 0, 2, 3, 1}, i, j, t;  
    for(i=1; i<5; i++) {  
        t=a[i]; j=i-1;  
        while(j>=0&& t>a[j]) {  
            a[j+1]=a[j];  
            a[j]=t;  
            j--;  
        }  
    }  
}
```

则该程序段的功能是（ ）

A 对数组a进行升序排序 B 对数组a进行降序排序 C 将数组中的每个元素依次左移一个位置 D 将数组中的每个元素依次右移一个位置 参考答案 B

两个数组 a 和 b 进行如下初始化：

```
char a[] = "abcde";
```

```
char b[] = {'a', 'b', 'c', 'd', 'e'};
```

则以下叙述正确的是（ ）。

A a 与 b 数组完全相同 B a 与 b 长度相同 C a 和 b 中都存放字符串 D a 数组比 b 数组长度长 参考答案 D

若实参是数组名，则实际传递给形参的是（ ）

A 数组的第一个元素值 B 数组首地址 C 数组的全部元素值 D 数组元素的个数 参考答案 B

若有

```
char x[] = "12345";
```

```
char y[] = { '1', '2', '3', '4', '5' };
```

则有（ ）。

A x数组和y数组长度相等 B x数组长度大于y数组长度 C x数组长度小于y数组长度 D

x数组等价于y数组

参考答案 B

若有二维数组 `int a[n][m]`；其中n、m是常量则`a[i][j]`之前的元素个数是（ ）

A $j*n+i$ B $i*m+j$ C $j*n+i-1$ D $i*m+j-1$ 参考答案 B

若有`int a[3][5]={2, 2}, {2, 6}, {2, 6, 2}}`，则数组a共有（ ）个元素。

A 7 B 9 C 6 D 15 参考答案 D

在定义一个数组时，如果没有给出数组元素的初始值，则所有元素的取值为0。 [参考答案] 错误

对于一个数组a，a与&a[0]都表示数组中首元素的存储地址，该地址称为数组的首地址。 [参考答案] 正确

在使用函数strlen计算一个字符串的长度时，不计算空格符的个数，但要计算''\0''的个数。 [参考答案] 错误

有如下程序段：

```
char C[20] = "examination";
```

```
C[4]=0;
```

```
cout<<c<<endl;
```

执行这个程序段的输出是__1__。

参考答案1: exam

假设用一个长度为50 的数组（数组元素的下标从0 到49）作为栈的存储空间，栈底指针bottom 指向栈底元素，栈顶指针top 指向栈顶元素，如果bottom=49， top=30（数组下标），则栈中具有__1__个元素。

参考答案1: 19

若已定义，以下fun函数的功能是在循环中给数组a前4个数组元素依次赋1、2、3、4；在第二个循环中使a数组前4个元素中的 值对称折叠，

变成1、2、2、1。请填空。 fun(int a[]) { int i;
for(i=1;i<=4;i++)__1__ = i; for(i=0;i<2;i++)__2__=a[i];

参考答案1: a[i-1]2: a[3-i]

阅读以下程序： #include<iostream.h> void abc(char str[]) {
int a,b; for(a=b=0;str[a]!='\0';a++) if(str[a]!='c')
str[b++]=str[a]; str[b]='\0'; } int main() { char
str[]="abcdef"; abc(str); cout<<"str[]="<<str<<endl;
return 0; } 程序运行时的输出结果是__1__。

参考答案1: str[]=abdef

已知下列程序先计算出杨辉三角数表存于二维数组a的对应位置各元素中，然后输出数组a主对角元素及以下的元素便得到如图 所示的杨辉三角数表的前几行。

```
1
11
121
1331
14641
#include<iostream>
#include<iomanip>
using namespace std;
void main()
{
    int i, j, a[5][5];
    for(i=0; i<5; i++)
        {a[i][0]=1; __1__

            for(j=1; __2__
j++) a[i][j]=a[i-1][j-1]+a[i-1][j];}
    for(i=0; i<5; i++)
        {cout<<"\n";
            for(j=0; __3__
; j++) cout<<setw(6)<<a[i][j];}
```

```
cout<<"\n";  
}
```

参考答案

1: a[i][i]=1;2: j<i ;或 i>j;3: j<=i; 或 i<=j;

数组的指针就是数组的__1__。

参考答案1: 数组名或首地址或起始地址或地址

C++中规定, 数组元素的下标从__1__开始。

参考答案1: 零 或 0

c++对数组存取不会进行下标__1__检查。

参考答案1: 越界

将n个数按从小到大顺序排列后输出。

```
#define N 10  
  
#include <iostream>  
  
#include <iomanip>  
  
using namespace std;  
  
int main( )  
{int i, j, t;  
int a[N];  
cout<<"please input 10 numbers:"<<endl;  
for (i=0;i<N;i++)  
cin>>a[i];  
for (i=0;__1__;i++)  
for (j=i+1;j<N;j++)
```

```

        if ( __2__ )
    { t=a[i];
        a[i]=a[j]; __3__ ;}
    cout<<"The sorted numbers:"<<' \n' ;
    for(i=0;i
        cout<return 0;

```

参考答案1: $i < N-1$ 或 $N-1 > i$ 2: $a[i] > a[j]$ 或 $a[j] < a[i]$ 3: $a[j] = t$

字符串 “\’a\’xy=4\n” 的长度为__1__.

参考答案1: 8

找出矩阵A 3×4 中第一个最大元素以及它的行号和列号(从0开始计算)。

```

#define M 3
#define N 4
#include <iostream>
using namespace std;
int main( )
{int i, j, row, colum, max;
int a[M][N];
for(i=0; i<M; i++)
    for(j=0; j<N; j++)
        cin>>a[i][j];
max=__1__; row=0, colum=0;
for (i=0; i<M; i++)

```

```

for(j=0;j<N;j++)
    if (__2__)
        { max=a[i][j];
          row=i;
          colum=j;
        }
cout<<" \nmax=" <<max<<" , row=" << row<<" ,
colum=" <<colum<<' \n' ;
return 0;
}

```

参考答案1: a[0][0]2: max<a[i][j]或a[i][j]>max

输入一批（10个）整数和一个指定的某整数，查找这批数中是否有该指定数，并得出相应结果：找到“Found”，没找到“Not Found”。

```

#include <iostream>
using namespace std;
int fun (int a[],int t,int x)
{ int i,f;
  f=0;
  for(i=0;i<t;i++)
      if (__1__)
          f=1;
  return f;
}
int main()

```

```

{  int number[10], i, x, y;
    cout<<"input 10 numbers:";
    for(i=0;i<10;i++)
        cin>>number[i];
    cout<<"input a special number:";
    cin>>x;
    y=fun ( __2__, 10, x);
    if(y)  cout<<"Found" <<endl;;
    else  cout<<"Not Found" <<endl;;
    return 0;
}

```

参考答案1: $a[i]==x$ 或 $x==a[i]$ 2: number

已知费氏 (Fibonacci) 数列通项:

$$F_0 = F_1 = 1 \quad (i=0, 1)$$

$$F_i = F_{i-1} + F_{i-2} \quad (i>1)$$

用数组求Fibonacci数列前20项。

```

#include <iostream>
#include <iomanip>
using namespace std;
int main( )
{  int  i, f[20];
  f[0]= f[1]=__1__;

```

```

    for(i=2;i<20;i++)
        f[i]=f[i-1]+__2__;
    cout<<" Fibonacci :"<<endl;
    for(i=0;i<20;i++)
        {if(i%5==0)    cout<<endl;
          cout<<setw(6)<<f[i];
        }
    return  0;
}

```

参考答案1: 12: f[i-2]

函数int lookup(int x[], int n,int y)在x[0]...x[n-1]中查找是否有等于y的元素，若有，返回第一个相等元素的下标，否则，返回-1，请将函数补充完整。

```

int lookup(int x[],int n,int y)
{int flag=-1;
int i;
for(i=0;i<__1__;i++)
if(x[i]==y)
    {flag=i;
    break;
    }
return  __2__;
}

```

```
}
```

参考答案1: n2: flag

下面程序的功能是删除字符串s中的空格。void main()

```
{
    char s[]="Beijing ligong daxue";
    int i, j;
    for(i=j=__1__;s[i]!='\0';i++)    if(s[i]!=' ')    s[j++]=__2__;
    s[j]='\0';
    cout<<s<<endl;
}
```

参考答案1: 02: s[i]

下面函数的功能是从数组x（有n>0个元素）中找出最大元素，请将函数补充完整。

```
int findmax(int x[],int n)
{ int max=__1__;
for(int i=1;i<n;++i)
    if(x[i]>max)
        __2__;
return max;
}
```

参考答案1: x[0]2: max=x[i]

已知下面程序先计算出杨辉三角数表存于二维数组a的对应位置各元素中，然后输出数组a主对角元素及以下的元素便得到如下 杨辉三角数表

的前几行，

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

请完善其空缺部分。

```
#include <iostream>
```

```
using namespace std;
```

```
#include <iomanip>
```

```
int main()
```

```
{ int i, j, a[5][5];
```

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

```
        {__1__; a[i][i]=1; for(j=1; __2__; j++) a[i][j]=a[i-1][j-1]+a[i-1][j]; }
```

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

```
{cout<<"\n";
```

```
for(j=0; __3__; j++) cout<<setw(6)<<a[i][j]; }
```

```
cout<<endl;
```

```
return 0;
```

```
}
```

参考答案1: a[i][0]=1; j<i或i>j; j<=i或i>=j

下面的程序用选择法对输入的N个数按从小到大顺序排序。

```

using namespace std;
#include <iostream.h>
#include <iomanip.h>
#define N 10
int main()
{int a[N], i, k, j, t;
for(i=0; i<N; i++) cin>>__1__;
for(i=0; i<N-1; i++)
    {k=i;
    for(j=i+1; j<N; j++)
        if( a[j]<__2__) k=j;    if(k!=i) {t=__3__; a[i]= a[k];
a[k]= t ; }
    }
for(i=0; i<N; i++)
    cout<<setw(6)<<a[i];

cout<<endl;
return 0;
}

```

参考答案1: a[i] 2: a[k] 3: a[i]

数组是具有相同__1__的一组数据构成的整体。 参考答案1: 类型

若有定义int a[3][4]={ {1, 2}, {3}, {4, 6, 8, 10}}, 则a[1][2]的值为__1__。 参考答案1: 0

要使用strcat、strlen等字符串处理函数，则需要预处理命令#include<__1__>。 参考答案1: string 或 string.h

以下程序运行后的输出结果是（ ）。

```
#include <iostream>

using namespace std;

int main()
{
    int i, a[][3]={1, 2, 3, 4, 5, 6, 7, 8, 9};
    for(i=0; i<3; i++) cout<<a[2-i][i];
    return 0;
}
```

__1__。

参考答案1: 753

```
#include <iostream>

using namespace std;

int main()
{
    char p[]={' a' , ' b' , ' c' }, q[]="abc";
    cout<<sizeof(p)<<' ', ''<<sizeof(q)<<endl;
    return 0;
}
```

上述程序运行后的输出结果是

__1__。

参考答案1: 3, 4

使用函数strcat(str1, str2)连接两个字符串时，连接的结果保存在字

符数组__1__中。 参考答案1: str1

设有以下定义语句: `int a[3][2]={10, 20, 30, 40, 50, 60};`

`int b[2][3]={10, 20, 30, 40, 50, 60};`

则`a[1][1]*b[1][1]=`

`__1__`。

参考答案1: 2000

下面程序的功能是: 将字符数组a中下标值为偶数的元素从小到大排列, 其他元素不变。请填空。

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
main() {
```

```
char a[]=" zhongnandaxue" ; //偶数个数
```

```
char t;
```

```
int i, j, k;
```

```
k=strlen(a);
```

```
for(i=0; i<=k-2; i+=2)
```

```
for(j=i+2; j<=k;
```

```
__1__)
```

```
if(
```

```
__2__) {
```

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

```
}
```

```
cout<<a<<endl;  
}
```

参考答案1: $j+=22$: $a[j]<a[i]$ 或 $a[i]>a[j]$

以下程序把 b 字符串连接到 a 字符串的后面。

```
void main() {  
    char a[20]="abcde ", b[]="fgh ";  
    int nm=0, n=0;  
    while(a[num] !=  
    __1__)  
        num++;  
    while(b[n]) {  
        a(num)=b[n];  
        num++;  
        __2__;  
    }  
    a[num]=' \0 ' ;  
}
```

参考答案1: ' \0 ' 2: $n++$ 或 $n=n+1$

设有以下定义的语句:

```
int a[3][2]={10, 20, 30, 40, 50, 60};
```

```
int b[3][2]={10, 20, 30, 40, 50, 60};
```

则 $a[1][1]*b[2][1]=$ __1__。

参考答案1: 2400

假定一个一维数组的定义为“char a[8],b;”,若把该数组最后一个元素赋值给b,采取的表达式是__1__。参考答案1: b=a[7];

以下程序中的select()函数功能是:在N行M列的二维数组中选出一个最大值作为函数值返回,并通过形参传回此最大值的行下标。请填空。

```
#include<iostream>

using namespace std;

#define N 3
#define M 3

int select(int a[N][M],int *n){
    int i,j,row=0,column=0;
    for(i=1;i<N;i++)
        for(j=0;j<M;j++)
            if(a[i][j]>a[row][column]){
                row=i;
                column=j;
            }
    *n=
    __1__;
    return
    __2__;
}

void main(){
    int a[N][M]={2 ,4, 1, 5, 3, 7, 6, 9, 8};
```

```

int max,n;
max=select(a,&n);
cout<<" max=" <<max<<" line=" <<n<<endl;
}

```

参考答案1: row2: a[row][column]

已知invert(int a[], int n)将数组a中的n个元素前后颠倒重新存放，请完善。

```

void invert(int a[], int n)
{
    int t,k,i;
    k= n/2;
    for( i=__1__; i<k; i++){ t= a[i] ; a[i]=a[n-1-i];
    a[n-1-i]=__2__; } }

```

参考答案1: 02: t

下面的函数 void sort(int b [],int n)对其形参数组b中的 n 个数用冒泡法按从小到大次序 排序。请填空。

```

void sort(int b [ ],int k )
{
    int i,j,t,flag;
    for (j=0;j<k-1;j++)
    {
        flag=0;
        for (i=0; i<
        __1__; i++)
        if (b[i]
        __2__b[i+1]) { t=b[i]; //相邻元素交换位置
            b[i]=b[i+1];

```

```

        b[i+1]=t;
        flag=1;//有元素交换位置，标志
    }

    if (flag == 0 ) break;  //没有交换元素，结束循环
}
}

```

参考答案1: $k-j-1$ 或 $k-j-2$: $>$ 或 $>=$

方矩阵 $A_{N \times N}$ 存于一个二维数组 $a[N][N]$ 中，现要求 A 的转置矩阵 A' ，且将结果仍然存于 a 中(即将 $a[i][j]$ 与 $a[j][i]$ 的值互换)。矩阵 A 从键盘输入，且输出 A' 到屏幕。

```

#include <iostream.h>
#include <iomanip.h>
#define N 5
//using namespace std;

int main( )
{
    int a[N][N], i, j, t;
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            cin >> a[i][j];
    for(i=0; i<N; i++)
        for(j=0; j<=__1__; j++)
            { t = a[j][i]; a[j][i] = a[i][j];
              a[i][j] = __2__; }
    for(i=0; i<N; i++)
        { for(j=0; j<N; j++)

```



```

cout<<setw(6)<<a[i][j];
cout<<' '\n';
}
return 0;
}

```

参考答案1: i 或 i-12: t

设数组a[N]的前N-1个元素的值已按从小到大排好序, 今再输入一个数, 将此数 插入 到合适位置, 使a[N] 中的 N 按 从小到大排 序。

```

#include <iostream.h>
#include <iomanip>
//using namespace std;
#define N 8
void main( )
{
    int a[N]={3,6,9,13,34,56, 78}; // 前N-1个元素值 已从 小到大
    排序
    int i, x;
    cout<<"Input x: ";
    cin>>x;           // 再 输入一个 新的数
    for(i=N-2;i>0;i--)
    {   if(a[i]>x)
        a[i+1]=a[i];
    else
        __1__
    }
}

```

```

}
a[__2__]=x;
for(i=0;i<8;i++)
cout<<a[i]<<' '\t';
cout<<endl;
}

```

参考答案1: break2: i+1 或 ++i

如下程序运行时输出为 a[4]=

```

#include <iostream>
using namespace std;
int main()
{int i, j, a[5]={1, 0, 0, 0, 0 };
for(j=0;j<5;j++)
for(i=0;i<j;++i)
a[j]=a[j]+a[i];
cout<<"a[4]="<<a[4]<<endl;
return 0;
}

```

参考答案 8

如下程序的输出为: a[m][n]= m= n=

```

#include <iostream>
using namespace std;
int main()
{ int i, j, m, n, a[3][4]={43, 32, 22, 13, 11, 22, 33, 44, 44, 11, 32, 31};

```

```

    m=n=0;
    for(i=0;i<3;i++)
        for(j=0;j<4;j++)
            if (a[m][n]>a[i][j]) {m=i;n=j;}
    cout<<"a[m][n]="<<a[m][n]<<", m="<<m<<", n="<<n<<endl;
    return 0;
}  参考答案 11, 1, 0

```

```

#include <iostream.h>

void main()
{
    int i;
    int a[3][3]={1,2,3,4,5,6,7,8,9};
    for(i=0;i<3;i++)
        cout<<a[2-i][i]<<' \t'
    cout <<endl;
}  参考答案 7 5 3

```

运行下列程序时，若输入

China

America

English

Japan

End

写出输出结果。

```

#include <iostream>

```

```

#include <string>
using namespace std;
void main()
{char str[100][20];
  int i,k,m;
  m=0;
  for (i=0;i<100;i++)
  {  cin>>str[i];
    if(strcmp(str[i],"End")==0)  break;
    k=strlen(str[i]);
    if(k>m) m=k;
  }
  cout<<m;
}  参考答案 7

```

当运行下面程序时，从键盘上输入3 4 2 6 1 5回车后得到的输出结果为

```

#include <iostream>
using namespace std;
void main()
{int a[6],i,j,k,m;
  for (i=0;i<6;i++)
    cin>>a[i];
  for (i=5;i>=0;i--)

```

```

{ k=a[5];
  for (j=4;j>=0;j--)
    a[j+1]=a[j];
  a[0]=k;
}
for (m=0;m<6;m++)
  cout<<a[m]<<" ";
cout<<endl;
} 参考答案 3 4 2 6 1 5

```

请写出下面程序运行的结果。

```

#include <iostream>
using namespace std;
void main()
{ int k,a[3][3]={1,2,3,4,5,6,7,8,9};
  for(k=0;k<3;k++)
    cout<<a[k][2-k]<<" ";
  cout<<endl;
} 参考答案 3 5 7

```

请写出下面程序运行的结果。

```

void main()
{int aa[3][3]={ {1,3,5}, {2,4,6}, {3,6,9} };
  int i,s=0;
  for(i=0;i<3;i++) s+=aa[i][1];
}

```

```
cout<<s<<endl;
```

```
} 参考答案 13
```

请写出下面程序运行的结果。

```
#include <iostream>
```

```
using namespace std;
```

```
int sum(int x[],int n)
```

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

```
for(i=0;i<n;i++) s+=x[i];
```

```
return s;}
```

```
int main( )
```

```
{int a[5]={1,2,3,4,5};
```

```
int s1,s2;
```

```
s1=sum(a,5);
```

```
s2=sum(a+2,3);
```

```
cout<<s1<<" "<<s2<<endl;
```

```
return 0;
```

```
} 参考答案 15 12
```

请写出下面程序运行的结果。

```
void main()
```

```
{ int i,j,a[5]={1};
```

```
for(i=1;i<5;i++)
```

```
for(j=i-1;j>=0;j--)
```

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

```
cout<<a[4];
```

```
} 参考答案 8
```

请写出下面程序运行的结果。

```
void main()
```

```
{char a[]="ab12cd34ef";
```

```
int i, j;
```

```
for(i=j=0;a[i];i++)
```

```
if(a[i]>='a' && a[i]<='z') a[j++]=a[i];
```

```
    a[j]='\0';
```

```
cout<<a<<endl;
```

```
} 参考答案 abcdef
```

请写出下面程序运行的结果。

```
void main()
```

```
{ int i, j, m, n, a[3][4]={43, 32, 22, 13, 11, 22, 33, 44,  
44, 11, 32, 31};
```

```
    m=n=0;
```

```
    for(i=0;i<3;i++)
```

```
for(j=0;j<4;j++)
```

```
if (a[m][n]>a[i][j]) {m=i;n=j;}
```

```
cout<< a[m][n]<<" "<< m<<" "<<n<<' \n';
```

```
}
```

运行时输出为： 参考答案 11 1 0

阅读程序，写出程序输出结果

```
#include <iostream.h>

int f( char s[],char t[] )
{ int i,j,k;
for(i=0; s[i]!=''\0';i++)
{ for(j=i,k=0; s[j]!=''\0'&& t[k]!=''\0'&& s[j]==t[k];
j++,k++);
if( t[k]==''\0') return i;
}
return -1;
}

int main( )
{char s1[100]="abcefg",*s2="ce";
cout<<"\n"<<f(s1,s2)<<"\n";
return 0;
}
```

输出结果() **参考答案 2**

阅读程序，写出程序输出结果

```
#include <iostream.h>

void main()
{
int num[3][3]={ {1,6,9}, {3,5,4}, {2,7,8} };
int i,j,it,itt, t,jt,exist=0;
```



```

for (i=0;i<3;++i)
{ t= num[i][0];
for (j=0;j<3;++j)
if ( t<num[i][j]) { t=num[i][j], jt=j; }
for (it=0;it<3;++it)
if ( t>num[it][jt]){ t=num[it][jt], itt=it; }
if (i==itt) { exist=1;break;}
}
if (exist) cout<<t <<endl;
else cout<<"no" <<endl;
}

```

输出结果(). 参考答案 5

阅读程序，写出程序输出结果

```

#include <iostream.h>

void main()
{
int num[5]={1,0,0,0,0};
int i,j;
for (j=0;j<5;++j)
for (i=0;i<j;++i)
num[j]=num[j]+num[i];
cout<<num[4] <<endl;
}

```

输出结果() 参考答案 8

程序中有如下语句: `for(int i=0;i<5;i++) cout<<*(p+i)<<"，";` 能够依次输出int型一维数组a的前5个元素。由此可知, 变量p的定义及初始化语句是()

A `int *p=a;` B `int p=a;` C `*p=a;` D `p=a;` 参考答案 A

若有以下定义和语句: `int s[4][5], (*p)[5]; p=s;` 则指针对s数组中第三个元素的正确引用形式是()。

A `p[0][3]` B `p[1][2]` C `p[0][2]` D `p[1][3]` 参考答案 C

下列程序的输出结果是()。 `#include<iostream> using namespace std; int main() { char a[]="Hello,Test"; char *p=a; while (*p) { if(*p>='a' && *p<='z') cout<<char(*p+'A'-'a'); else cout<<*p; p++; } return 0; }`

A `hello, test` B `Hello, Test` C `HELLO, TEST` D `hELLO, tEST` 参考答案 C

有下列程序: `#include<iostream.h> void main() { int x; int &y=x; y=99; cout<<"x="<<x<<endl; }` 则程序的运行结果是()。

A `x=99` B `x=9` C `x=0` D 异常 参考答案 A

执行语句序列: `int x=1, &y=x; cout<<x<<'-'<<y<<endl;` 输出的结果是()。

A `x-x` B `1-1` C `1-0` D 异常 参考答案 B

下列关于指针的运算中, () 是非法的。

A 两个指针在一定条件下, 可以进行相等或不等的比较运算 B 可以将一个空指针赋值给某个指针 C 一个指针可以加上两个整数之差 D 两个指针在一定条件下可以相加 参考答案 D

下列关于指针的用法中, 错误的是()。 A `int i; int *p=&i;` B `int i; int *p; i=*p;` C `int *p; p=0;` D `int i=5;`

`int *p; p=&i;` 参考答案 B

有以下定义及初始化: `int a=3, *p=&a;` 则由此可以推断, `*p`的值是 ()。

A 变量a的值, 即3 B 变量a的地址值 C 变量p的地址值 D 无意义 参考答案 A

有如下定义: `int a[5]={1, 3, 5, 7, 9}, *p=a;` 下列表达式中不能得到数值5的是 ()。

A `a[2]` B `a[3]` C `*(p+2)` D `*p+4` 参考答案 A

语句 `int *p=&k;` 定义了指针p, 与这个语句等效的语句序列是 ()。

A `int *p; p=&k;` B `int *p; p=k;` C `int *p; *p=&k;` D `int *p; *p=k;` 参考答案 A

有如下程序: `#include<iostream> using namespace std; int main() { int *p; *p=9; cout<<"The value at p:"<<*p; return 0; }`

编译运行程序时将出现的情况是 ()。

A 编译时出现语法错误, 不能生成可执行文件 B 运行时一定输出The value at p:9 C 运行时一定输出The value at p: *9 D 运行时有可能出错 参考答案 D

语句`int *p=&a;`, 与这个语句等效的是_____。

A `int *p;p=&a;` B `int *p;*p=&a;` C `int *p;p=a;` D `int *p;*p=a;` 参考答案 A

若已定义:

`int a[]={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, *p=a, i;`

其中 $0 \leq i \leq 9$, 则对a数组元素不正确的引用是_____。

A `a[p-a]` B `*(&a[i])` C `p[i]` D `a[10]` 参考答案 D

有关指针变量说法不正确的是_____。

A 指针变量有地址 B 指针变量是一个不变的地址数 C 指针变量是一个4字节存储单元 D 指针变量可存放存储单元地址数 **参考答案 B**

对于int *pa[5]的描述, ()是正确的。

A pa是一个指向数组的指针, 所指向的数组是5个int型元素 B pa是一个指向某数组中的第5个元素的指针, 该元素是int型变量 C pa[5]表示某个数组的第5个元素的值 D pa是一个具有5个元素的指针数组, 每个元素是一个int型指针 **参考答案 D**

已知

```
char c;
```

```
char * const pc="asdf";
```

()是正确的。

A pc="dfgh" B *pc='b' C *pc++='y' D pc=&c **参考答案 B**

有二维字符数组char s[4][6]={"zhang", "gao", "tang", "wang"}, 执行程序cout<<*s[2], 屏幕上显示 ()。 A gao B tang C g D t **参考答案 B**

已知 char s[10]=" abcdefg", *p=s; 下面不允许出现的语句是 ()。

A p=s+5; B s=p+1; C s[2]=p[4]; D *p=s[3]; **参考答案 B**

定义语句 int (*p)(); 的含义是 ()。

A p是一个函数名, 该函数返回值是指针 B p是一个指向一维数组的指针变量 C p是一个指向函数的指针, 该函数返回值是整数 D 以上都不对 **参考答案 C**

变量s的定义为char *s="Hello world!"; , 要使变量p指向s所指向的字符串, 则应选取 ()。

A char *p=s; B char *p=&s; C char *p;p=*s; D char *p; p=&s; **参考答案 A**

要使指针变量p指向一维数组a的第1个元素(下标为0的元素), 正确的

赋值表达式是()。

A $p=a$ 或 $p=a[0]$ B $p=a$ 或 $p=\&a[0]$ C $p=\&a$ 或 $p=a[0]$ D $p=\&a$ 或 $p=\&a[0]$ 参考答案 B

下列表达式中，与下标引用 $a[k]$ 等效的是()。

A $a+k$ B $*a+k$ C $*(a+k)$ D $a+*k$ 参考答案 C

设有 $\text{int } a[10]$, $*p=a$; 则下面哪种表示与 $a[5]$ 不等价 ()。

A $*(a+5)$ B $*(p+5)$ C $p[5]$ D $p+5$ 参考答案 D

有 $\text{int } i, a[10]$, $*p=a$; 则下面()不是对 a 数组元素的正确引用，其中 $0 \leq i \leq 9$ 。

A $a[p-a]$ B $*(\&a[i])$ C $p[i]$ D $*(*(a+i))$ 参考答案 D

设有 $\text{int } a[10]$, $*p=a$; 则下面哪条语句有错误 ()。

A $p=p+1$; B $p[0]=*p+1$; C $a[0]=a[0]+1$; D $a=a+1$; 参考答案 D

对于基类型相同的指针变量，下面进行下面哪种运算是没有意义的 ()。

A $+ B - C = D =$ 参考答案 A

已知一运行正常的程序中有这样两个语句：

```
int *p1,*p2=&a;
```

```
p1=b;
```

由此可推知，变量 a 和 b 的类型分别是 ()。 A int 和 int B int 和 $\text{int} *$ C $\text{int} *$ 和 int D $\text{int} *$ 和 $\text{int} *$ 参考答案 B

若有： $\text{int } a, *p=\&a$; 则下面哪个表达式为假 ()。

A $*p= \&a$ B $p= \&*p$ C $p= \&a$ D $*p= a$ 参考答案 A

设有 $\text{int } a[20]$, $*p=x$; 则下面中哪个与 $x[1]$ 不等价 ()。 A $p[1]$ B $*++p$ C $*++a$ D $*(a+1)$ 参考答案 A

下面哪个定义或语句序列能使p指向a, ()。

A int a,*p=a; B int a,*p; *p=a; C int a,*p=&a; D int a,*p;
*p=&a; 参考答案 C

sizeof关键字的作用是_____。

A 求指针 B 返回变量的值 C 指定变量的值大小 D 求类型变量的字节数 参考答案 D

与delete关系最密的关键字是_____。 A namespace B new C enum
D struct 参考答案 B

假定变量a和pa定义为 double a[10], *pa=a; 要将60赋值给a中的下标为5的元素, 不正确的语句是 ()。 A pa[5]=60; B *
(pa+5)=60; C a[5]=60; D *(a[0]+5)=60; 参考答案 D

设有 int a[10], *p=&a[5] ; 则下面哪种表示与a[9]不等价:

A *(a+9) B *(p+4) C p[4] D p+4 参考答案 D

以下程序的运行结果是 ()。

```
#include <iostream>

using namespace std;

int main()
{
    int a=10,*b=&a,**c=&b;
    cout<<**c<<endl;
    return 0;
}
```

A 10 B 变量a的地址 C 变量b的地址 D 编译错误 参考答案 A

若有定义:

```
int a[]={1,2,3,4,5,6,7,8},*p=a;
```

则表达式 $*(p+1)$ 的值是 ()。

A 1 B 2 C 3 D 4 参考答案 B

下列有关指针的用法中错误的是 ()。

A `int i;int* Ptr=&i;` B `int i;int *Ptr;*Ptr=i;`

C `int* Ptr;Ptr=0;` D `int i=5;int* Ptr;Ptr=&i;` 参考答案 B

设有以下程序段:

```
int a[5]={0},*p,*q;
```

```
p=a;
```

```
q=a;
```

则下列选项中,合法的运算是 ()。

A $p+q$ B $p-q$ C $p*q$ D $p\%q$ 参考答案 B

有如下程序段,请指出**p所代表的是 ()。

```
int **p,*s,k=20;
```

```
s=&k;
```

```
p=&s;
```

A s B &k C *p D k 参考答案 D

若有以下定义“`int (*p)[4];`”,则标识符p是 ()。

A 一个指向整型变量的指针 B 一个指针数组名 C 一个指针,它指向一个含有4个整型元素的一维数组 D 说明不合法 参考答案 C

下列语句错误的是 ()。

A `char *p="John";p[2]='a';` B `char name[5]="John";`

`name[2]='a';` C `char name[5]="John",*p=name;p[2]='a';`

D `char name[5]=" John" , *p=&name[2]; *p=' 'a' ;` 参考答案 A

若函数形参是指针变量，则对应的实参 ().

A 只能是指针 B 只能是数组名 C 可以是指针或数组名 D 任何类型的数据 参考答案 C

对使用关键字new所开辟的动态存储空间，释放时必须使用 ()

A free B create C delete D realese 参考答案 C

若有 `int *p1,*p2,a[10]; p1=&a[1]; p2=&a[6];` 则 ()

A p1的值大于p2的值 B p1的值小于p2的值 C p1、p2没有可比性 D p1、p2哪个大不确定 参考答案 B

对于一个二维数组a，`a+i=a[i]=*(a+i)=&a[i]=&a[i][0]` 。

[参考答案] 正确

若有定义 `int (*p)[4];` ，则定义了一个包含4个指针值的一维数组。

[参考答案] 错误

阅读以下程序： `#include<iostream.h> #include<string.h> int fun(char *w, int m) { char s,*p1,*p2; p1=w; p2=w+m-1; while(p1<p2) { s=*p1++; *p1=*p2--; *p2=s; } return 0; } int main() { char a[]="ABCDEFGF"; fun(a,strlen(a)); cout<<a<<endl; return 0; }` 程序运行时的输出结果是__1__。

参考答案

1: AGAAGAG

语句 `int a,*p;` 定义可存放变量p地址的变量pp的语句是__1__。 参考答案1: `int **pp=&p;`

一个变量的引用变量名是该变量的__1__。

参考答案1: 别名

下面的程序用选择法对输入的10个数按从小到大顺序排序。


```

#include <iostream>
using namespace std;
int main( )
{
    int a[10], *p=a, i, k, j, t, n;
    n=__1__ ;
    for(i=0; i<n; i++) cin>>p[i];
    for(i=0; i<n-1; i++)
    {
        k=i;
        for(j=i+1; j<n; j++)
            if (__2__ ) k=j;

        if(k!=i) { t=a[i]; a[i] = a[k]; a[k]= __3__ ; }
    }
    for(p=a; p<a+n; p++) cout<<" "<<*p;
    cout<<endl;
    return 0;
}

```

参考答案1: 102: a[i]<a[j]或a[j]>a[i]3: t

以下程序中实现两个数的交换，请将程序补充完整。

```

#include<iostream>
using namespace std;
void main( )

```

```

{
    int a=5,b=9;
    void swap(int *, int *);
    swap(__1__ );
    cout<<"a="<<a<<"b="<<b<<endl;
}

void swap(int *p, int *q)
{
    __2__;
    t=*p; *p=*q; *q=t;
}

```

参考答案1: &a,&b2: int t;或int t

下面的程序求用户输入数据的整数部分和小数部分，如输入9.45，则输出：整数部分：9，小数部分：0.45。

```

void f(float y,int *a,float *z)
    {*a=(int)y ; *z=__1__-*a;}

void main()
{ int b;
  float x,w;
  cin>>x;
  f(x,&b,__2__);
  cout<<"整数部分："<<b<<"，"<<"小数部分："<<w<<endl;
}

```

参考答案1: y2: &w

有以下定义和语句 `int a[3][2]={1, 2, 3, 4, 5, 6}, *p[3]; p[0]=a[1];` 则输出 `*(p[0]+1)` 的值为__1__。 参考答案1: 4

已知: `int a[5], *p=a;` 则 `p` 指向数组元素是__1__。

参考答案1: a[0]

若有 `int *p1, *p2, a[10]; p1=&a[1]; p2=&a[6];` 则 `p1` 的值__1__ `p2` 的值 (空格处填大于、小于或等于)。

参考答案1: 小于

指针与数组名两者的区别主要在于: 两者中, __1__ 的值不能发生改变。 参考答案1: 数组名

数组作为函数的参数, 从实参到形参传递的是值还是地址? 答: 传递的是__1__。 参考答案1: 地址

设有 `char *s[4]={ "Jerry", "TOM", "Mary", "peter" };` 则 `strlen(s[3])` 的值为__1__。 参考答案1: 5

设有 `int *p, a=2, b=1;` 则执行以下语句 `p=&a; *p=a+b;` 后, `a` 的值为__1__。 参考答案1: 3

假设已有 `char *c= " \nab\0c\0"`, 则 `strlen(c)` 的返回值为__1__。 参考答案1: 3

编译系统用__1__代表函数的入口地址。 参考答1: 函数名

设有 `int a[3][2] = {1, 2, 3, 4, 5, 6}, (*p)[2]=a1;` 则 `*(*(p+2)+1)` 的值是__1__。 参考答案1: 6

若有定义:

`int a[]={1, 2, 3, 4, 5, 6, 7, 8}, *p=a;`

则 表达式 `*p++` 的值是

__1__。 参考答案1: 1

若有定义:

```
int a[]={1,2,3,4,5,6,7,8},*p=a;
```

则 表达式 *p+3 的值是

__1__。

参考答案1: 4

设有int a[3][2] = {1, 2, 3, 4, 5, 6}, (*p)[2]=a1;则*(*(p+2)+1)的值是__1__。 参考答案1: 6

设P是指向具有10个元素的一维数组的指针变量，该数组元素类型为int，则指针变量P的定义为__1__ 参考答案1: int (*P)[10];

若有“int a[4]={10,11,12,13}, b, *p=a;”，则执行语句“b=*p++;”后，b的值是__1__。接着执行语句“b=*++p;”后，b的值是 __2__。 参考答案1: 102: 12

” int &p;” 含义是定义一个整型变量的__1__。 参考答案1: 引用

下面的函数用来判断s指向的字符串是否为“回文串”，即从前向后或从后向前读是一样的(默认长度为0、1的字符串是“回文 串”)。

```
int f(char *s)
{
    int result=1;
    char *p=s;
    while(*p) p++;
    __1__;
    while(s<p)
    {
        if( *s!=*p) { result=0; break ; }
        __2__; p--; }
    return result;
}
```

参考答案1: p--或--p或p=p-12: s++或++s或s=s+1

函数int *lookup(int t[],int val,int n)在x[0]...x[n-1]中查找是否有等于val的元素，若有，返回第一个相等元素的指针，否则，返回 空指针，请将函数补充完整。

```
int *lookup(int t[],int val,int n)
{ int i,*j;
j=
__1__;
for(i=0;i<n;i++)
    if (t[i]==val) { j=__2__; break;}
return j;
}
```

参考答案1: NULL 或 02: j=&t[i] 或 j=t+i

设有int k[]={1,2,3,4,5,6,7},*p=k+2; 则*p的值为__1__。

参考答案1: 3

阅读程序，严格按输出顺序与格式写出程序运行结果

```
#include <iostream.h>
int Psfa(int *a,int b)
{ *a=*a+2;
  b=b+2;
  return (*a)*b;
}
void main()
{ int x=1,y=2,s;
  s=Psfa(&x,y);
```

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

```
cout<<"s"<<s;
```

```
} 参考答案 x=3y=2 s=12
```

```
#include<iostream>
```

```
using namespace std;
```

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

```
void fun(int *pa,int n);
```

```
void main( )
```

```
{ int m=8;
```

```
fun(a,m);
```

```
cout<<a[7]<<endl; }
```

```
void fun(int *pa,int n)
```

```
{ for (int i=0;i<n-1;i++)
```

```
*(pa+7)+= *(pa+i);} 参考答案 28
```

```
#include<iostream>
```

```
using namespace std;
```

```
#include <cstring >
```

```
void main( )
```

```
{char s1[20]="I love ", s2[]="China!",cmp;
```

```
cout<<"s1的长度是"<<strlen(s1) <<' \t';
```

```
int nc=strcmp(s1,s2);
```

```
switch (nc)
```

```
{ case 0: cmp='=';break;
```

```

        case 1: cmp='>';break;
        case -1: cmp='<';}
cout<< strcat(s1,s2) <<' \t' ;
cout
<<"s1"<<cmp<<"s2"<<endl;    }

```

参考答案 s1的长度是6 I love China! s1>s2

```

#include <iostream.h>
int main( )
{
    void sort(char *name[ ], int n);
    void print(char *name[ ], int n);
    char *name[ ]={ "BASIC", "FORTRAN", "C++", "PASCAL",
"COBOL" };
    int n=5;
    sort(name, n);
    print(name, n);
}
void sort(char *name[ ], int n)
{
    char *temp;
    int i, j, k;
    for(i=0; i<n-1; i++)
    { k=i;

```

```

        for(j=i+1; j<n; j++)
            if ( strcmp(name[k], name[j])>0) k=j;
            if(k!=i)
                { temp=name[i]; name[i]=name[k]; name[k]=temp; }
    }
}

```

```

void print(char *name[ ], int n)

```

```

{   int i;
    for( i=0; i<n; i++ )
        cout<<name[i]<<" ";
    cout<<endl;
}

```

} **参考答案** BASIC C++ COBOL FORTRAN PASCAL

请写出下面程序运行的结果。

```

#include <iostream>

using namespace std;

int  sub(char *s)
{char *p=s;
while(*p)  p++;
return (p-s);
}

void  main()
{char a[]="ABCDEFGH";
    cout<<sub(a)<<endl;
}

```


} 参考答案 7

请写出下面程序运行的结果。

```
#include <iostream>
using namespace std;
void swap(int *p1,int *p2)
{ int t;
  t=*p1;
  *p1=*p2;
  *p2=t;
}

int main()
{int a=3,b=5;
  swap(&a,&b);
  cout<<a<<" "<<b<<endl;
  return 0;
```

} 参考答案 5 3

请写出下面程序运行的结果

```
void f(char * st,int i)
{ st[i]='\0';
  cout<<st;
  if (i>1) f(st,i-1);
}

void main()
```

```
{ char st[]="abc";
    f(st,3);

    cout<<endl;
```

} **参考答案** abcbaba

阅读程序，写出程序输出结果()。

```
#include <iostream.h>

void f( char *p1,char *p2 )
{ while(*p1!=',' ',' ',' ',' ',' ',' ',' '\0',' ',' ',' ',' ',' ',' ',' '\0') p1++;
while(*p2!=',' ',' ',' ',' ',' ',' ',' '\0',' ',' ',' ',' ',' ',' '\0')
{ *p1=*p2;
p1++; p2++;}
*p1=',',' ',' ',' ',' ',' ',' '\0',' ',' ',' ',' ',' ',' '\0';
}

int main( )
{char s1[100]="ABC",*s2="A";
f(s1,s2);
cout<<',' ',' ',' ',' ',' ',' ',' '\n',' ',' ',' ',' ',' ',' '\n'<<s1<<',' ',' ',' ',' ',' ',' '\n',' ',' ',' ',' ',' ',' '\n';
return 0;
}
```

参考答案 ABCA

当定义一个结构体变量时，系统分配给它的内存大小的理论值是_____。

A 各成员所需内存的总和 B 结构体中的第一个成员所需内存 C 成员中所需内存最大者的存储空间 D 结构体中的最后一个成员所需存储容量

参考答案 A

存放多个字符串，用_____比较适合。

A 二维字符数组 B 一维字符数组 C 自定义结构体 D 自定义共用体 参

考答案 A

定义生成非静态变量时，将对变量进行_____。

A自动赋初值 B自动赋0 C不会自动赋初值 D自动赋-1

参考答案 C

以下定义，

```
struct st1{int a,b;float x,y};
```

```
struct st2{int a,b;st1 s1; } ss;
```

对成员变量x的引用形式正确的是：_____。 A ss.s1.x

B st2.st1.x C st2.s1.x D ss.x 参考答案 A

对任意整型数据m,n($m>n>0$)，则 C 语言表达式 $m-m/n*n$ 的值为：
_____。

A 0 B $m\%n$ C 1 D $n\%m$ 参考答案 B

设有以下定义：

```
enum t1 {a1,a2=7,a3,a4=15} time;
```

则枚举常量a1和a3的值分别是

A 1和2 B 6和3 C 1和8 D 0和8 参考答案 D

设有类型说明：enum color{red,yellow=3,white,black};

则执行语句cout<<white<<' \n' ;后输出的结果是_____。

A 4 B 3 C 2 D 1 参考答案 A

以下对枚举类型名的定义中正确的是 ()。 A enum a={“one”, “two”, “three”}; B enum a {“one”, “two”, “three”}; C enum a={one, two, three}; D enum a

{one=9, two=-1, three}; 参考答案 D

下面程序的正确的输出是（ ）。

```
#include <iostream.h>

void main()
{ enum team{my, your=4, his, her=his+10};
  cout<<my<<' ' <<your<<' ' <<his<<' ' <<her<<endl;
}
```

A 0123 B 04010 C 04 515 D 1 4 5 15 参考答案 C

下面程序的正确的输出是（ ）。

```
#include <iostream.h>

void main()
{ struct ex
{int x,y; };
ex num[2]={1, 3, 2, 7};
  cout<<num[0].y/num[0].x*num[1].x<<endl;
}
```

A 0 B 1 C 3 D 6 参考答案 D

设有以下说明语句

```
typedef struct
{ int n;
char ch[8];
}PER;
```

A PER是结构体变量名 B PER是结构体类型名 C typedef struct是结构

体类型 D struct是结构体类型名 参考答案 B

对于结构体变量，下列说法正确的是（ ）

```
struct st1{int a, b; float x, y;}s1, s2;
```

```
struct st2{int a, b; float x, y;}s3, s4;
```

A s1、s2、s3、s4可以相互赋值 B 只有s1和s2、s3和s4之间可以相互赋值 C s1、s2、s3、s4之间均不可以相互赋值 D 结构体变量不可以整体赋值 参考答案 B

某结构体变量定义如下，对此结构体变量的成员的引用形式正确的是（ ）

```
struct st{int a,b; float x,y;} s1,*p;
```

```
p=&s1;
```

A s1->a B p->b C p.x D *p.y 参考答案 B

double a,&a的数据类型为_____。

A int* B double C double * D unsigned int 参考答案 C

int a;*p=&a;, *a的数据类型为_____。 A int* B double C double * D int 参考答案 D

int a;double b;, a+4/b结果的数据类型为_____。

A int B double C float D unsigned 参考答案 B

char a;int b;, a+4/b结果的数据类型为_____。

A int B char C float D unsigned 参考答案 A

union关键字的含义是 _____。

A自动变量定义 B外部变量定义 C共用体类型定义 D枚举类型定义 参考答案 C

关键字typedef的作用是_____。

A 定义宏标识名 B 定义变量 C 给已有的类型名取别名 D 定义类名 参

考答案 C

已知枚举类型定义语句为:

```
enum Token
```

```
{ NAME, NUMBER, PLUS=5, MINUS, PRINT=10 };
```

则下列叙述中错误的是()。

A 枚举常量NAME的值为1 B 枚举常量NUMBER的值为1 C 枚举常量MINUS的值为6 D 枚举常量PRINT的值为10 考答案 A

已知教师记录的描述为

```
struct teacher{
```

```
int id;
```

```
char name[20];
```

```
struct { int y; int m; int d;} birth;
```

```
}t;
```

将变量t 中的 d 成员赋值为 12 的语句为()。

A d=12 B birth.d=12 C t.d=12 D t.birth.d=12 考答案 D

以下对共用体类型数据的正确叙述是 ()。

A 一旦定义了一个共用体变量后,即可引用该变量或该变量中的任意成员 B 一个共用体变量中可以同时存放其所有成员 C 一个共用体变量中不可以同时存放其所有成员 D 共用体类型数据可以出现在结构体类型定义中,但结构体类型定义数据不能出现在共用体类型定义中 考答案 C

typedef long int Big 的作用是 ()。 A 定义了一个整型变量 B 定义了一个长整型变量 C 建立了一种新的数据类型 D 定义了一个新的数据类型标识符 考答案 D

某结构体变量定义如下,结构体变量的成员的引用形式正确的是 ()。

```
struct str{ int m,n; float a,b; } s,*pt;
```

pt=&s;

A s->m B pt.n C *pt.a D pt->b 参考答案 D

下列语句中错误的是（ ）。

A const int a; B const int a=10; C const int *point=0;
D const int *point=new(10); 参考答案 A

结构体数组的每个数组元素都是一个结构体类型的数据。

[参考答案] 正确

在C++中，函数之间传递结构体变量的方法只有二种：一种是以结构体变量作为参数，直接传递结构体变量的值；第二种是以指向结构体指针作为参数，传递结构体变量的地址。 [参考答案] 错误

在C++中，函数之间传递结构体变量，可以用结构体变量的引用变量作为参数。 [参考答案] 正确

在C++中，数组的大小必须在编译时确定，而运算符new是用来进行动态内存分配的。 [参考答案] 正确

在C++中，delete 运算符只能释放用运算符 new 申请的动态内存空间。 [参考答案] 正确

自定义类型包括结构体、共用体和__1__三种类型。

参考答案1：枚举型或 枚举类型或 枚举体 或 枚举

从键盘输入不多于100个学生的数据，包括学号，姓名，性别等，统计输入的学生人数及男女学生人数（学号为0，则结束输入）。

```
#include<iostream>
```

```
using namespace std;
```

```
const int N=100;
```

```
struct student
```

```
{int num;
```

```
char name[10];
```

```

char sex;
}stu[N];
void main()
{int i=0,male=0,female=0,sum=0;
cout<<"输入学生信息!\n";
while(i<N)
{    cout<<"学号: ";
    cin>>stu[i].num;
    if(stu[i].num==0) __1__ ;cout<<"姓名: ";
    cin>>stu[i].name;
    cout<<"性别(f代表女, m代表男):";
    cin>>stu[i].sex;
    if(stu[i].sex=='m') male++;
    else female++;
    i++;}
cout<<"共输入"<< __2__<<"个学生的信息\n";
cout<<"其中男生人数为: "<<male;
cout<<"女生人数为: "<<female<<endl;}

```

参考答案1: break2: male+female 或 i

共用体变量中，变量的地址和它的各成员地址是同一个地址；因此，在任一时刻只能保存__1__个成员。

参考答案1: 一 或 壹 或 1

执行 typedef int ABC[10]; 语句把ABC定义为具有__1__个整型元素的数组类型。

参考答案1: 十 或 10

数组中的每一个元素必须是同一种数据类型，结构体中允许将__1__类型的数据组合成一种数据结构。参考答案1: 不同

一个结构体变量所占用的空间是各成员所需内存空间的__1__。参考答案1: 和 或 总和

指向结构体数组的指针的类型是__1__的类型。参考答案1: 指针

设有定义: enum color {red=5, yellow=0, blue=3, white, green};

则white的取值为__1__。参考答案1: 4

基本的浮点类型关键字有__1__和double 参考答案1: float

变量的定义格式为__1__

参考答案1: [存储类别] 数据类型 变量名表;

字符和字符串常数的定界符分别为__1__和”。

参考答案1: 单引号 或 ’ ’ 或 ’

定义整型变量的关键字为unsigned、__1__、__2__和__3__。

参考答案1: int2: long3: char

设有如下定义:

```
struct sk
```

```
{    int a;
```

```
    float b;
```

```
}data,*p=&data;
```

使用p为data的a成员赋值为5，正确的赋值语句是__1__。

参考答案1: p->a=5 或 (*p).a=5

有如下定义:

```
union exx{    int a;
```

```
char ch;  
struct {  
    int c;  
    char d; } s;  
} u = {20}, *p;
```

执行下列语句：

```
p = &u;  
u.ch = p->a + 77;  
u.s.c = p->ch - 32;  
u.s.d = (*p).ch + 32;  
cout << (*p).s.c << (*p).ch << p->a << p->s.d;
```

输出结果为

__1__。

参考答案1： 65A65a

有如下定义：

```
struct st { int x; int *y; } *p;  
int s[] = {5, 6, 7, 8};  
st a[] = {10, &s[0], 20, &s[1], 30, &s[2], 40, &s[3]};
```

执行下列语句：

```
p = a;  
cout << p->x;  
cout << (++p)->x;  
cout << *(++p)->y;
```

```
cout<<+>(*(&p)->y);
```

输出结果为（ ）。

__1__。

参考答案1: 102079

下面程序的功能是：以结构体变量表示复数，分别计算两个已给复数之积，并以复数形式输出结果。将程序补充完整。复数的乘法按照以下的法则进行：设 $z_1=a+bi$ ， $z_2=c+di$ ($a、b、c、d \in \mathbb{R}$) 是任意两个复数，那么它们的积 $(a+bi)(c+di)=(ac-bd)+(bc+ad)i$ 。

```
#include <iostream>

using namespace std;

struct complex{
    double re,im;
};

void printcomp(complex);

viod main() {
    complex z1={2.0, 5.0}, z2={4.0, 7.0}, zmu;
    zmu.re=
    __1__ zmu.im=__2__ cout<<"z1=";    printcomp(z1);    cout<<endl;
    cout<<"z2=";    printcomp(z2);    cout<<endl;
    cout<<"z1*z2=";
    __3__ cout<<endl;
}

void printcomp(complex z)
{
```

```

if(z.im>0)
cout<<z.re<<'+'<<z.im<<'i';
else
cout<<z.re<<'-'<<z.im<<'i';
}

```

参考答案1: $z1.re*z2.re-z1.im*z2.im$; **或** $z2.re*z1.re-z2.im*z1.im$; $2z1.re*z2.im+z1.im*z2.re$; **或** $z2.im*z1.re+z2.re*z1.im$;
 printcomp(zmu);

设有定义如下:

```

union ex{
short int i;
char ch;
};
ex data;

```

执行下列语句:

```

data.i=0x5566;
cout<<hex<<data.i<<endl;

```

输出结果为

__1__继续执行下列语句:

```

data.ch=' A' ;
cout<<data.ch<<endl;

```

输出结果为

__2__继续执行下列语句:

```

cout<<hex<<data.i<<endl;

```

输出结果为

__3__。

参考答案1: 55662: A3: 5541

有定义如下:

```
struct per
{
    int i;
    char c;
    char *cp;
};
per q={62, ' f' ,''Luo''};
void func(per &a)
{
    a.i=59;
    a.c=' m' ;
    a.cp=''Li'';
}
```

执行下列语句:

```
func(q);
cout<<q.i<<q.c<<q.cp;
```

输出结果为

__1__。

参考答案1: 59mLi

动态分配内存使用关键字new，释放内存使用关键字__1__。 参考答案
1: delete

动态分配内存使用关键字__1__，释放内存使用关键字delete。 参考答
案1: new

```
#include<iostream.h>

union ex
{
    short int i;
    char ch;
};

void main()
{
    ex data;
    data.i=0x5566;
    cout<<"data.i="<<hex<<data.i<<' , ' ;
    data.ch='A' ;
    cout<<"data.ch="<<data.ch<<' , ' ;
    cout<<"data.i="<<hex<<data.i<<endl;
}  参考答案 data.i=5566, data.ch=A, data.i=5541
```

有如下类定义:

```
class A
{
    char *a;
public:
    A( ) : a(0) {}
    A(char *aa)
```

```

    {    //把aa所指字符串拷贝到a所指向的存储空间
        a=_____ ;
        strcpy(a, aa);
    }
    ~A( ) (delete []a; }
};

```

横线处应填写的表达式是()。 A new char[strlen(aa)+1]
 B char[strlen(aa)+1] C char[strlen(aa)] D new
 char[sizeof(aa)-1] 参考答案 A

有如下程序：

```

# include <iostream>
using namespace std;
class Name
{
    char name[20];
public:
    Name( )
    {    strcpy(name, ""); cout<<'?' ;    }
    Name(char *fname)
    {    strcpy(name, fname); cout<<'?' ;    }
};
int main( )
{
    Name names[3]={Name("张三"), Name("李四")};

```

```
        return 0;
    }
}
```

运行此程序输出符号?的个数是()。

A 0 B 1 C 2 D 3 参考答案 D

有如下程序:

```
# include <iostream>

class Toy
{
public:
    Toy(char* _n)
    {   strcpy(name, _n);   count++;   }
    ~Toy( ) {   count--;   }
    char* GetName( )
    {   return name;   }
    static int getCount( )
    {   return count;   }
private:
    char name[10];
    static int count;
};

int Toy::count=0;

int main( )
{
```



```
Toy t1("Snoopy"), t2("Mickey"), t3("Barbie");  
cout<<t1.getCount( )<<endl;  
return 0;  
}
```

运行时的输出结果是()。

A 1 B 2 C 3 D 运行时出错 参考答案 C

有如下两个类定义：

```
class AA{};  
class BB  
{  
    AA v1,*v2;  
    BB v3;  
    int *v4;  
};
```

其中有一个成员变量的定义是错误的，这个变量是()。 A v1
B v2 C v3 D v4 参考答案 C

有如下程序：

```
#include<iostream>  
using namespace std;  
class AA  
{  
    int k;  
protected:
```

```

        int n;
        void setK(int k){ this->k=k; }
public:
        void setN(int n){ this->n=n; }
};
class BB: public AA
{ /*类体略*/ };
int main( )
{
    BB x;
    x.n=1;                //1#
    x.setN(2);            //2#
    x.k=3;                //3#
    x.setK(4);            //4#
    return 0;
}

```

在标注号码的四条语句中正确的是()。

A 1 B 2 C 3 D 4 参考答案 B

有如下程序：

```

#include<iostream.h>
#include<string.h>
class XCF
{

```

```

        int a;
public:
    XCF (int aa=0) : a(aa) { cout<<"1"; }
    XCF(XCF & x)
    {
        a=x.a;
        cout<<"2";
    }
    ~XCF( ) { cout<<a; }
    int Geta( ) { return a; }
};

void main( )
{
    XCF d1(5), d2(d1);
    XCF *pd=new XCF(8);
    cout<<pd->Geta( );
    delete pd;
}

```

运行时的输出结果是（ ）。

A 1215588 B 1218855 C 12185 D 128512 **参考答案 B**

建立一个类对象时，系统自动调用

A 析构函数 B 构造函数 C 静态函数 D 友元函数 **参考答案 B**

一个类的友元函数能够访问该类的_____。

A 私有成员 B 保护成员 C 公有成员 D 所有成员 参考答案 D

C++中的模块包括_____。

A 对象模块和函数模块 B 对象模块和类模块 C 函数模块和类模块
D 对象模块和变量模块 参考答案 C

下列有关类的说法，不正确的是_____。

A 类是一种用户自定义的数据类型 B 只有类中的成员函数或类的友元函数才能存取类中的私有数据 C 在类中，如果不作特别说明，所有的数据成员均为私有数据成员 D 在类中，如果不作特别说明，所有的成员均为公有成员 参考答案 D

下列关于构造函数说法不正确的是_____。

A 构造函数必须与类同名 B 构造函数可以省略不写 C 构造函数必须有返回值 D 在构造函数中可以对类中的成员进行初始化 参考答案 C

下列关于C++语言类的描述中错误的是_____。

A 类用于描述事物的属性和对事物的操作 B 类与类之间通过封装而具有明确的独立性 C 类与类之间必须是平等的关系，而不能组成层次结构 D 类与类之间可以通过一些方法进行通信和联络 参考答案 C

在面向对象程序设计中，实现信息隐蔽是靠_____。

A 对象的继承 B 对象的封装 C 对象的分类 D 对象的多态 参考答案 B

有关类的多态性不正确的说法是_____。

A 同名函数成员可有不同的参数和类型 B 同一个操作符可表示不同的处理 C 同类对象的类定义 D 模板编程 参考答案 C

有关类与对象的说法不正确的是_____。

A 类是同类对象的定义 B 对象是类的具体化实例 C 类包含对象的具体属性值 D 类包含同类对象的属性名 参考答案 C

下面有关类与对象的说法中，错误的是_____。

A 对象是类的实例 B 一个对象只能属于一个具体的类 C 一个类只能有

一个对象 D 类和对象的关系与数据类型和变量的关系相似 参考答案 C
类的概念是()。

A 对象的抽象。 B 同一类记录的集合。 C 同一类数据的集合。 D 所有对象的集合。 参考答案 A

设有以下类和对象的定义：

```
class A
{public:
int k;
};
```

A x1, x2, *p1, *p2;

则下面针对成员变量k的引用，正确的是()。

A x1->k=1; B x2.k=2; C p1.k=3; D (*p2)->k=4; 参考答案 B

设有以下类的定义：

```
class Ex
{ int x;
public:
void setx(int t=0);
};
```

若在类外定义成员函数setx(), 以下定义形式中正确的是()。

A void setx(int t) {...} B void Ex::setx(int t) {...}

C Ex::void setx(int t) {...} D void Ex::setx() {...} 参考答案 B

假设ClassA为一个类，则该类的复制初始化构造函数的声明语句为()。

A ClassA(ClassA p); B ClassA &(ClassA p); C ClassA(ClassA

&p);D ClassA (ClassA *p); 参考答案 C

下面对构造函数和析构函数的定义，正确的是（ ）。

A void A::A(), void A::~A() B A::A(参数), A::~A()C A::A(参数), A::~A(参数)D void A::A(参数), void A::~A(参数) 参考答案 B

如果没有显式定义构造函数（包括复制构造函数），C++编译器就（ ）。

A 出现编译错误B 没有构造函数C 必须显示定义

D 隐式定义默认的构造函数 参考答案 D

为了使类中的某个成员不能被类的对象通过成员操作符访问，则不能把该成员的访问权限定义为（ ）。

A public B protected C privateD static 参考答案 A

下列各类函数中，不是类的成员函数的是（ ）。

A构造函数B复制初始化构造函数 C析构函数D友元函数

参考答案 D

关于成员函数特征的下述描述中，错误的是（ ）。

A 成员函数一定是内联函数B 成员函数可以重载C 成员函数可以设置参数的默认值D 成员函数可以是静态的 参考答案 A

对类的构造函数和析构函数描述正确的是（ ）。

A 构造函数可以重载，析构函数不能重载B 构造函数不能重载，析构函数可以重载C 构造函数和析构函数均可以重载 D 构造函数和析构函数均不能重载 参考答案 A

以下有关析构函数的叙述，不正确的是（ ）。

A 在一个类只能定义一个析构函数 B 析构函数和构造函数一样可以有形参 C 析构函数不允许用返回值D 析构函数名前必须冠有符号“~”

参考答案 B

下面关于友元函数描述正确的是（ ）。

A 友元函数的实现必须在类的内部定义 B 友元函数是类的成员 C 友元函数破坏了类的封装性和隐藏性 D 友元函数不能访问类的私有成员 **参考答案 C**

下面对静态数据成员的描述中，正确的是（ ）。

A 类的每一个对象都有自己的静态数据成员 B 类的不同对象有不同的静态数据成员值 C 静态数据成员不能通过类的对象调用 D 静态数据成员是类的所有对象共享的数据 **参考答案 D**

以下不是构造函数特征的是（ ）。

A 构造函数的函数名与类名相同 B 构造函数可以重载 C 构造函数可以设置默认参数 D 构造函数必须指定类型说明 **参考答案 D**

假定MyClass为一个类，那么下列的函数说明中，（ ）为该类的析构函数。

A void ~MyClass(); B ~MyClass(int n); C MyClass(); D ~MyClass(); **参考答案 D**

有如下类声明：

```
class TestClass
{
    int i;
private:
    int j;
protected:
    int k;
public:
    int m,n;
};
```

其中，私有成员的数量为（ ）。

A 0 B 1 C 2 D 3 参考答案 C

下列有关类的说法，不正确的是（ ）。

A 对象是类的一个实例 B 任何一个对象只能属于一个具体的类 C 一个类只能有一个对象 D 类与对象的关系和数据类型与变量的关系相似 参考答案 C

在C++中，编译系统自动为一个类生成缺省构造函数的条件是（ ）。

A 该类没有定义任何有参构造函数 B 该类没有定义任何无参构造函数
C 该类没有定义任何构造函数 D 该类没有定义任何成员函数 参考答案 C

关于友元，下列说法正确的是（ ）。 A a 类为 b 类的友元类，则 a 类中的所有函数就可以访问 b 类中的所有成员 B 友元函数的定义在类体中 C 友元只能是类

D 友元只能是函数 参考答案 A

设已定义类 SS，下列关于该类的常对象定义语句中，错误的是（ ）。

A const SS s; B SS const s; C SS const s(0); D

SS s(0) const;

参考答案 D

以下程序运行后，输出星号（*）的个数是（ ）。

```
#include <iostream>
```

```
using namespace std;
```

```
class Sample
```

```
{
```

```
public:
```

```
Sample() { }
```



```

~Sample() {cout<<' '*';}

};

int main() {
Sample temp[2],*pTemp[2];
return 0;
}

```

A 1 B 2 C 3 D 4 参考答案 B

下列程序运行后的输出结果是()。

```

#include <iostream.h>

class Sample{
char c1,c2;
public:
Sample(char a)    {c2=(c1=a)-32;}
void disp()      { cout<<c1<<"转换为"<<c2<<endl; }
};

void main() {
Sample a(''a''),b(''b'');
a.disp();
b.disp();
}

```

A a转换为Ab转换为B B a转换为Bb转换为AC a转换为A b转换为BD a转换为B b转换为A 参考答案 A

在面向对象程序设计中，实现信息隐蔽是靠（ ）。

A 对象的继承 B 对象的封装 C 对象的分类 D 对象的多态 参考答案 B

下列关于构造函数说法不正确的是（ ）。

A构造函数必须与类同名B构造函数可以省略不写 C构造函数必须有返回值D在构造函数中可以对类中的成员进行初始化 参考答案 C

下列有关类的说法，不正确的是（ ）。

A 类是一种用户自定义的数据类型B 只有类中的成员函数或类的友元函数才能存取类中的私有数据C 在类中，如果不作特别说明，所有的数据均为私有数据D 在类中，如果不作特别说明，所有的成员函数均为公有数据 参考答案 D

一个类的友元函数能够访问该类的是（ ）。

A 私有成员B 保护成员 C 公有成员 D 所有成员 参考答案 D

有关类与对象的说法不正确的是（ ）。

A 类是同类对象的定义B 对象是类的具体化实例 C 类包含对象的具体属性值D 类包含同类对象的属性名 参考答案 C

对类成员访问权限的控制，是通过设置成员的访问控制属性实现的，下列不是访问控制属性的是（ ）。

A 公有类型B 私有类型 C 保护类型 D 友元类型 参考答案 D

在类的定义中，用于为对象分配内存空间，对类的数据成员进行初始化并执行其他内部管理操作的函数是（ ）。

A 友元函数B 虚函数 C 构造函数 D 析构函数 参考答案 C

下述静态数据成员的特征中，（ ）是错误的。

A 说明静态数据成员时前边要加修饰符static B 静态数据成员要在类体外进行初始化C 引用静态数据成员时，要在静态数据成员名前加<类名>和作用域运算符D 静态数据成员不是所有对象所共用的 参考答案 D

下列有关类的说法不正确的是（ ）。

A 对象是类的一个实例B 任何一个对象只能属于一个具体的类

C 一个类只能有一个对象 D 类与对象的关系和数据类型与变量的关系相似 **参考答案 C**

为了使类中的某个成员不能被类的对象通过成员操作符访问，则不能把该成员的访问权限定义为（ ）。

A public B protected C private D static **参考答案 A**

（ ）是类的所用对象共享的成员，而不是某个对象的成员。

A 静态成员 B 动态成员 C 内联函数 D 析构函数 **参考答案 A**

下面类的定义，有（ ）处错误。

```
class MyClass{  
private:  
int i=0;  
public:  
void MyClass();  
~MyClass(int Value);  
};
```

A 1 B 2 C 3 D 4 **参考答案 A**

下列关于析构函数的描述中，错误的是（ ）。

A 类中有且仅有一个析构函数 B 析构函数可以有形参 C 析构函数没有函数返回类型 D 析构函数在对象消失时被自动执行 **参考答案 B**

若有一个MyClass类，则执行语句“`MyClass a, b(2), *p;`”后，自动调用该类的构造函数（ ）次。

A 1 B 2 C 3 D 4 **参考答案 B**

下列对类的构造函数和析构函数的描述中，正确的是

A 构造函数可以重载，析构函数不能重载 B 构造函数不能重载，析构函数可以重载 C 构造函数可以重载，析构函数也可以重载 D 构造函数

不能重载，析构函数也不能重载 参考答案 A

面向对象程序设计将数据与()放在一起，作为一个相互依存、不可分割的整体来处理。

A .对数据的操作 B 信息 C 数据隐藏 D数据抽象 参考答案 A

下面对于常数据成员的描述正确的是()

A 常数据成员可以不初始化，并且不能更新。 B 常数据成员必须初始化，并且不能更新。 C 常数据成员可以不初始化，并且可以被更新。 D 常数据成员必须初始化，并且可以更新。 参考答案 B

下面对于析构函数的描述错误的是()

A 类中有且仅有一个析构函数。 B 析构函数可以有形参。 C 析构函数没有函数类型。 D 析构函数在对象消失时被自动执行。 参考答案 B

若要把函数void f()定义为类aClass的友元函数，则应该在类aClass定义中加入的语句是()。

A void f() B static void f() C friend f() D friend void f()
参考答案 D

若有一个MyClass类，则执行语句“`MyClass a,b(2),*p;`”后，自动调用该类的构造函数()次。

A 2 B 3 C 4 D 5 参考答案 A

下列关于类的权限的描述错误的是 A 类本身的成员函数只能访问自身的私有成员

B 类的对象只能访问该类的公有成员 C 普通函数不能直接访问类的公有成员，必须通过对象访问 D 一个类可以将另一个类的对象作为成员
参考答案 A

如果没有为一个类定义任何构造函数的情况下，下列描述正确的是

A 编译器总是自动创建一个不带参数的构造函数 B 这个类没有构造函数 C 这个类不需要构造函数 D 该类不能通过编译 参考答案 A

const是常函数类型的组成部分，在函数实现时不需要有此关键字。

[参考答案] 错误

静态成员变量的初始化只能在类体外进行。 [参考答案] 正确
任何成员函数都可以对类的常对象和常数据成员进行操作。

[参考答案] 错误

类是一种数据结构，系统会给类分配存储空间。 [参考答案] 错误

this指针存储的值是当前被调用的成员函数所在的对象的地址。 [参考答案] 正确

构造函数可以是内联函数 [参考答案] 正确

公有继承是基类中的private成员在派生类中仍是private的

[参考答案] 错误

有如下类定义：

```
class Sample{  
public:  
    Sample();  
    ~Sample();  
private:  
    static int data;  
};
```

将静态数据成员data初始化为0的语句是__1__。

参考答案1: int Sample::data = 0;

在c++类的定义中，利用__1__描述对象的特征，利用__2__描述对象的行为。

参考答案1: 属性或数据成员2: 函数成员或方法 或 成员函数

面向对象的模型中，最基本的概念是对象和__1__。 参考答案1: 类

完成下列Point类定义中的析构函数，其中应删除pname指向的存储单

元。

```
class Point
{
    int xVal, yVal;
    char * pname;
public:
    Point( ) { pname=new char[10]; }
    Point(int x,int y) { xVal=x; yVal=y; }
    __1__
    { __2__ }
    void show( ) { cout<<"x="<<xVal<<"y="<<yVal;}
};
```

参考答案1: ~Point()2: delete []pname;

已知有类定义:

```
class A
{
    int a,b;
public:
    void f(int x, int y)
    {
        a=x;
        b=y;
    }
```

```
};
```

在Visual C++6.0 中sizeof(A)值为：__1__。

参考答案1： 8

在面向过程的程序设计中，程序功能模块由函数实现；在面向对象的程序设计中，程序模块由__1__实现。参考答案1： 类

在C++中定义类时，通常用__1__描述对象的特征，利用成员函数（方法）描述对象的行为或功能。参考答案1： 数据成员

找出并输出数组中的最大值。

```
#include<iostream>

using namespace std;

class Sample
{
int x;
public:
void setx(int x0){x=x0;}
friend int fun(Sample x[],int n)
{
int m= __1__ ;
for (int i=1;i<n;i++)
if (m<x[i].x)m=x[i].x;
return m;
}
};

void main()
```

```

{
int i;
Sample a[10];
int array[]={23, 56, 2, 5, 9, 10, 67, -12, 3, 45};
for(i=0;i<10;i++)
a[i].setx(array[i]);
cout<<__2__<<endl;
}

```

参考答案1: x[0].x2: fun(a, 10)

已知main函数中有A类对象ab，程序的运行结果为12，请完成以下代码：

```

#include<iostream.h>

class A
{
public:
int x,y;
void display() {cout<<x<<y;}
};

void main( )
{
__1__ ;
ab.x=1;
__2__ ;
}

```



```
ab.display( );  
}
```

参考答案1: A ab 或 A ab;2: ab.y=2

构造类型有__1__、结构体和公用体。参考答案1: 数组

当一个对象定义时，C++编译系统自动调用__1__建立该对象并进行初始化。参考答案1: 构造函数

如果不做特殊说明，类成员的默认访问权限是__1__

参考答案1: 私有 或 private

当一个对象的生命周期结束时，C++编译系统自动调用__1__撤销该对象并进行清理工作。参考答案1: 析构函数

设有如下程序结构：

```
class Box  
{...};  
  
void main()  
{Box A,B,C; }
```

该程序运行时调用__1__次构造函数。

参考答案1: 3

在C++类的定义中，利用__1__描述对象的特征。

参考答案1: 属性 或 数据成员

在C++类的定义中，利用__1__描述对象的行为。

参考答案1: 成员函数

类是对象的抽象，而对象则是类的具体__1__。参考答案1: 实例

类的成员按访问权限可分为3类，分别是public、private、__1__。参考答案1: protected

在面向对象的程序设计中，将数据和处理数据的操作封装成一个整体就定义了一种事物的类型，称作“类”。类是一种抽象的概念，属于该类的一个实例叫做“__1__”。**参考答案1：**对象

OOP技术由对象、__1__、方法、消息和继承五个基本的概念所组成。**参考答案1：**类

类是用户定义的类型，具有类类型的变量称作__1__。

参考答案1：对象

以下程序的功能是：找出数组中的最小值并输出，补充程序。

```
#include<iostream.h>

class sample
{ int x;
public:
    void setx( int x0) {x=x0;}
    friend int fun( sample b[ ], int n)
    { int m=__1__;
      for(int i=0; i<n;i++)
          if (b[i].x<m) m=__2__;
      return m;
    }
};

void main()
{sample a[6];
  int arr[]={12,6,21,7,10,9};
  for( int i=0;i<6;i++)
```

```
    a[i].setx(arr[i]);  
cout<<fun(__3__)<<endl;}
```

参考答案1: b[0].x或b[0].x;2: b[i].x或b[i].x;3: a,6

类的成员包括数据成员和__1__成员。参考答案1: 函数
使程序执行结果为100。

```
#include <iostream.h>  
  
class Test  
{  
public:  
    Test(int aa)  
    {__1__;  
    }  
    int GetX()  
    {__2__;    }  
private:  
    int X;  
};  
  
void main()  
{  
    Test xx(100);  
    cout<<xx.GetX()<<endl;  
}
```

参考答案1: X=aa2: return X

若一个3位整数的各位数字的立方和等于1099，则称该数为A数，求全部A数之和。

```
#include<iostream.h>

class number
{ private:
    int a,b;
public:
    void mn(int m,int n)
    {a=m; b=n;}
    void print()
    { int m1 = 0, m2 = 0, m3 = 0, m = 0, sum=0;
for(m = a; m <= b; m++)
    { m1 = __1__;
      m2 = (m / 10) % 10;
      m3 = m/100;
      if (m1*m1*m1 + m2*m2*m2+ m3*m3*m3 == 1099) sum+=m;
    }
    cout<<sum<<endl;
}

};

void main()
{ number ob;__2__
  ob.print();
}
```

```
}
```

参考答案1: `m % 102: ob.mn(100, 999);`

已知 $y=f(40)/[f(30)+f(20)]$, 当 $f(n)=1\times 2+2\times 3+3\times 4+\cdots+n\times (n+1)$ 时, 求 y 的值。

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

```
class calculate
```

```
{ public:
```

```
    float f(int n)
```

```
    { int i;
```

```
        float sum = 0;
```

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

```
            sum += i * (i + 1);
```

```
        return sum;
```

```
    }
```

```
void print()
```

```
{ float y = 0.0;
```

```
    y = __1__;
```

```
    cout<<y<<endl;
```

```
}
```

```
};
```

```
void main()
```

```
{ calculate ob;__2__ }
```

参考答案1: `f(40)/ (f(30) + f(20))2: ob.print();`

C++中还有一种特殊的指针，用来指向对象自己，这个指针就是__1__指针。参考答案1: this

面向对象程序设计的三个特性是封装性、继承性和__1__。

参考答案1: 多态性

在类的对象被创建时，__1__函数被自动调用。参考答案1: 构造函数
用于给对象的数据成员进行初始化的函数是

__1__。参考答案1: 构造函数

静态数据成员定义时，应在数据类型前面加上关键字__1__。参考答案1: static

以下程序的功能是：设计一个Employee类，包括编号、姓名和工资等私有数据成员，不含任何成员函数，只将main() 设置为该 类的友元函数，在主函数中输出编号、姓名和工资等数据。

```
#include <iostream.h>

class Employee
{
    int no;
    char name[10];
    float salary;
public: __1__ void main();
};

void main()
{ __2__
    cin>>obj.no>>obj.name>>obj.salary;

    cout<<obj.name<<"的编号是"<<obj.no<<"，工资为"
    <<obj.salary<<endl;
```

```
}
```

参考答案1: friend2: Employee obj;

面向对象的模型中，最基本的概念是对象和__1__。参考答案1: 类

继承的方式有公有继承、私有继承和__1__继承。参考答案1: 保护

在c++类的定义中，利用数据成员描述对象的特征，利用__1__描述对象的行为。参考答案1: 成员函数

在面向过程的程序设计中，程序功能模块由函数实现；在面向对象的程序设计中，程序模块由__1__实现。参考答案1: 类

抽象的主要原则是过程抽象和__1__抽象。参考答案1: 数据

关键字__1__写在函数参数列表之后，函数体之前，说明该函数是常成员函数。参考答案1: const

如果要把返回值为void的无参函数A声明为类B的友元函数，则应在类B的定义中加入的语句是__1__。参考答案1: friend void A();

当一个对象定义时，C++编译系统自动调用__1__建立该对象并进行初始化。参考答案1: 构造函数

设A为Population类的对象且赋有初值，则语句Population B(A);表示__1__。参考答案1: 将对象A复制给对象B

利用“对象名.成员变量”形式访问的对象成员仅限于被声明为public的成员。若要访问其他成员变量，需要通过成员函数或__1__函数。参考答案1: 友元

如果不做特殊说明，类成员的默认访问权限是__1__。

参考答案1: private 或 私有

成员对象的初始化必须通过构造函数的__1__进行。

参考答案1: 成员初始化列表

下列程序在构造函数和析构函数中申请和释放类的私有成员，请完成该类的实现。

```
class MyClass{
public:
MyClass(int a);
~MyClass(int a);
private:
int *x;
};

MyClass::MyClass(int a) {
__1__;
}

MyClass::~~MyClass() {
__2__;
}
```

参考答案1: x=new int(a)2: delete x

友元函数不是当前类的__1__，而是独立于当前类的__2__。

参考答案1: 成员函数2: 外部函数

在未指定访问权限时，结构中的成员被默认为__1__，而类中的成员被默认为__2__。**参考答案1:** 公有成员2: 私有成员

非成员函数应声明为类的__1__才能访问这个类的private成员。

参考答案1: 友元函数

常对象在定义时必须进行__1__。**参考答案1:** 初始化

对象数组的数组名是__1__。**参考答案1:** 常量指针

常数据成员的初始化只能通过构造函数的__1__。

参考答案1：成员初始化列表

一般来说，静态成员函数是访问类中的__1__。

参考答案1：静态数据成员

已知长方形的类R，main函数中定义R类对象r，其长为5，宽为3，完成以下代码，计算其面积和周长。

```
#include<iostream>

using namespace std;

class R
{
    int a,b;
public:
    R(int x,int y){a=x;b=y;}
    int area(){return a*b;}
    int perimeter(){return 2*a+2*b;}
};

int main()
{
    R __1__;
    cout<<"面积："<<__2__<<" 周长："<<r.perimeter()<<endl;
    return 0;
}
```

参考答案1：r(3,5) 或 r(5,3) 2：r.area()

在面向对象方法中，类之间共享属性和操作的机制称为 __1__.

参考答案1：继承

```
#include<iostream.h>

class number
{
private:
    int val;
public:
    number(int i) {val=i;}
    operator int( );
};

number::operator int( )
{
    //cout<<"operator int()..."<<val<<endl;
    return val;
}

void main( )
{
    number n(15);                //调用有一个参数的构造
    函数
    int i=n;                      //调用operator
    int() 类对象赋值给整型变量
    cout<<i<<endl;
    i+=n;                         //调用operator
    int()
```

```

        cout<<i<<endl;
        cout<<int(n)<<endl;           //调用operator int()
}  参考答案 15 30 15

```

阅读程序，严格按输出顺序与格式写出程序运行结果。

```

#include <iostream.h>

class Point
{
    int xVal, yVal;
public:
    Point( ) { cout<<"no value set"<<endl; }
    Point(int x,int y) { xVal=x;yVal=y; }
    void show( ) { cout<<"x="<<xVal<<"y="<<yVal<<endl; }
};

void main( )
{
    Point a, * p=new Point;
    Point b(1,2);
    *p=b;
    (*p).show( );
}

```

参考答案 no value set no value set x=1y=2

如下程序的运行结果是：

```

#include <iostream>

```

```

using namespace std;
class Base
{
    public:
    virtual void print() {cout<<"B";}
};
class Derived:public Base
{
    public:
    void print() {cout<<"D";}
};
int main()
{
    Base *pb;
    Derived d, *pd=&d;;
    pb=pd;
    pb->print();
    pd->print();
    cout<<endl;
    return 0;
}

```

参考答案 DD

如下程序的运行结果是：

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

```

class A
{
    int a;
public:
    A(int aa):a(aa) { cout<<a<<','; }
    set_a(int x) { a=x; }
    ~A( );
};
A::~~A( ){ cout<<a<<endl; }
void main( )
{
    A x(2);
    x.set_a(10);
}

```

参考答案 2,10

```

#include<iostream>
#include<iomanip>
using namespace std;
class A
{
public:
    A( );
    A(int i, int j);

```

```

        ~A( ) { cout<<"Destructor.\n"; }
    void print( );
private:
        int a,b;
};
A::A( )
{
    a=5; b=10;
    cout<<"Default constructor.\t"; }
A::A(int i, int j)
{ a=i,b=j;cout<<"Constructor.\t"; }
void A::print( )
{ cout<<"a="<<a<<"",b="<<b<<' \t' ; }
void main( )
{
    A obj1, obj2(15, 18);
    obj1.print( );
    obj2.print( );
}

```

参考答案 Default constructor. Constructor. a=5,b=10 a=15,b=18
Destructor. Destructor.

```

#include <iostream.h>

class myclass
{ int a,b;

```

```

public:
void init(int i, int j)
{a=i; b=j;}
friend int sum(myclass x);
};

int sum( myclass x)
{return x.a+x.b; }

void main()
{ myclass y;
y.init(37,41);
cout<<sum(y)<<endl;
}  参考答案 78

#include <iostream.h>

class Count
{
public:
Count() { count++;}
static int HM() {return count;}
~Count()
{ count--;}
private:
static int count;
};

```

```

int Count::count=10;
void main()
{
Count c1, c2, c3, c4;
cout<<Count::HM()<<' , ' <<c1.HM()<<endl;
} 参考答案 14, 14
#include <iostream.h>
class A
{
public:
A(double t, double r)
{ Total=t; Rate=r;}
friend double Count(A &a)
{
a.Total +=a.Rate*a.Total;
return a.Total;
}
private:
double Total, Rate;
};
void main()
{ A a1(100.0, 0.35), a2(200.0, 0.02);
cout<<Count(a1)<<' , ' <<Count(a2)<<endl;

```


} 参考答案 135,204

```
#include <iostream.h>

class Sample
{
    int x;
public:
    Sample() {} ;
    void setx(int i) {x=i;}
    friend int fun(Sample B[],int n)
    { int m=0;
      for (int i=0;i<n;i++)
        if (B[i].x>m) m=B[i].x;
      return m;
    }
};

void main()
{ Sample A[10];
  int Arr[]={90,87,42,78,97,84,60,55,78,65};
  for(int i=0;i<10;i++)
    A[i].setx(Arr[i]);
  cout<<fun(A,10)<<endl;
}
```

} 参考答案 97

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

```

class Sample
{private:
int i;
static int k;
public:
Sample();
void Show();
};

int Sample::k=0;
Sample::Sample()
{i=0; k++;}
void Sample::Show()
{cout<<"i="<<i<<" ,k="<<k<<endl;}

void main()
{ Sample a;
a.Show();
Sample b;
b.Show();
} 参考答案 i=0, k=1 i=0, k=2

#include <iostream.h>

int count=0;

class Point
{ int x,y;

```

```

public:
    Point()
    { x=1;y=1;
      count++;
    }
    ~Point()
    {count--;}
    friend void play();
};

void play()
{ count++;}

void main()
{
    Point a;
    play();
    {
        Point b[5];
        play();
    }
    play();
    cout << count << endl;
} 参考答案 4

#include <iostream.h>

```

```

class A
{ public:
int x;
A(int i) {x=i;}
void fun1(int j)
{ x+=j;
cout<<"fun1:"<<x<<endl;
}
void fun2(int j)
{ x+=j;
cout<<"fun2:"<<x<<endl;
}
};

void main()
{ A c1(2), c2(5);
void (A::*pfun)(int)=A::fun1;
(c1.*pfun)(5);
pfun=A::fun2;
(c2.*pfun)(10);
} 参考答案 fun1:7 fun2:15

#include <iostream.h>

class Sample
{

```

```

    char c1,c2;
public:
    Sample(char a) {c2=(c1=a)-32;}
    void disp()
    {
        cout<<c1<<"转换为"<<c2<<endl;
    }
};
void main()
{
    Sample a('a');
    a.disp();
} 参考答案 a转换为A

```

有如下程序：

```

#include<iostream>
using namespace std;
class VAC
{
public:
    int f( ) const { return 3; }
    int f( ) { return 5; }
};
int main( )

```

```

{
    VAC v1;

    const VAC v2;

    cout<<v1.f()<<v2.f();

    return 0;

}

```

写出其运行时的输出结果。 参考答案 53

有如下程序：#include <iostream.h>

```

class Test
{
public:
    Test( )    { n+=2; }
    ~Test( )   { n-=3; }
    static int getNum( ) { return n; }
private:
    static int n;
};

int Test::n=1;

void main( )
{
    Test * p=new Test;

    delete p;

    cout<<"n="<<Test::getNum( )<<endl;
}

```

```
}
```

写出执行该程序的输出结果。 参考答案 n=0

有如下程序：

```
#include<iostream.h>

class Con
{
char ID;
public:
Con() {ID=' 'A' ';cout<<1;}
Con(char id) {ID=id;cout<<2;}
char getID() {return ID;}
};

int main()
{
Con c1;
cout<<c1.getID();
Con c2(' 'B' ');
cout<<c2.getID();
return 0;
}
```

写出此程序的运行结果。

参考答案 1A2B

下列关于运算符重载的描述中，错误的是()

A ::运算符不能重载 B 类型转换运算符只能作为成员函数重载 C 将运算符作为非成员函数重载时必须定义为友元 D 重载[]运算符应完成“下标访问”操作 **参考答案 C**

将运算符重载为类成员函数时，其参数表中没有参数，说明该运算符是

A 不合法的运算符 B 一元运算符 C 无操作数的运算符 D 二元运算符
参考答案 B

下列关于运算符函数的描述中，错误的是()。

A 运算符函数的名称总是以operator为前缀 B 运算符函数的参数可以是对象 C 运算符函数只能定义为类的成员函数 D 在表达式中使用重载的运算符相当于调用运算符重载函数 **参考答案 C**

有如下程序：

```
#include<iostream>
```

```
class Pair
```

```
{
```

```
    int m;
```

```
    int n;
```

```
public:
```

```
    Pair (int i, int j): m(i), n(j) {}
```

```
    bool operator > (Pair p) const;           //须在类体外给出定
```

义

```
};
```

```
void main( )
```

```
{
```

```
    Pair p1(3,4), p2(4,3), p3(4,5);
```

```
    cout<<(p1>p2)<<(p2>p1)<<(p2>p3)<<(p3>p2);
```



```
}
```

运算符函数`operator>`的功能是比较两个`Pair` 对象的大小，当左边对象大时，返回`true`，否则返回`false`。比较规则是首先比较 两对象的`m` 成员，`m` 大者为大；当`m` 相等时比较`n`，`n` 大者为大。程序输出0101，下列对运算符重载函数的正确定义是（ ）。 A `bool`

```
Pair::operator>(Pair p) const { if (m!=p.m) return m>p.m; return n>p.n; }
```

```
 B bool Pair::operator>(Pair p) { if (m!=p.m) return m>p.m;
```

```
return n>p.n; } C bool Pair::operator>(Pair p) const
```

```
{ if (m>p.m) return true; return n>p.n; } D bool
```

```
Pair::operator>(Pair p) { if (m>p.m) return true; return n>p.n; } 参考答案 A
```

已知表达式`++a` 中的“`++`”是作为成员函数重载的运算符，则与`++a` 等效的运算符函数调用形式为 A `.operator++(1)` B `operator++(a)`

C `operator++(a,1)` D `a.operator++()` 参考答案 D

运算符重载时不需要保持的性质是

A 操作数个数 B 操作数类型 C 优先级 D 结合性 参考答案 B

一个函数为`void f(int, char ch='a')`，另一个函数`void f(int)`，则它们（ ）。

A 不能在同一程序中定义 B 可以在同一程序中定义并可重载 C 可以在同一程序中定义，但不可以重载 D 以上三种说法均不正确 参考答案 B

下面有关重载的说法中，错误的是 _____。

A 函数重载要求同名函数在参数个数或参数类型上不同 B 运算符重载是用同一个运算符针对不同类型数据进行不同的运算操作 C 所有的运算符都可以重载 D 运算符重载函数通常是类的成员函数和友元函数 参考答案 C

下面的函数声明中，_____是” `void BC(int a, int b);`” 的重载函数？

A int BC(int a, int b) B void BC(int c, int d) C float
BC(int a, int b, int c = 0) D void BC(int a, int b=0) 参考答案 C

有如下的运算符重载函数定义：

```
double operator +(int i , int k)
{ return double(i+k); }
```

但定义有错误，对这个错误最准确的描述是()。

A +只能作为成员函数重载，而这里的+是作为非成员函数重载的 B 两个int型参数的和也应该是int型，而这里将+的返回类型声明为double
C 没有将运算符重载函数声明为某个类的友元 D C++已经提供了求两个int型数据之和的运算符+，不能再定义同样的运算符 参考答案 D

下列对重载函数的描述中，错误的是()

A 重载函数中不允许使用默认参数 B 重载函数中编译是根据参数表进行选择
C 不要使用重载函数来描述毫不相干的函数 D 构造函数重载将会给初始化带来多种方式 参考答案 A

下面有关重载函数的描述中，正确的是()。

A 重载函数必须具有不同的返回值类型 B 重载函数形参个数必须不同
C 重载函数必须具有不同的形参列表 D 重载函数名可以不同 参考答案 C

关于运算符重载，下列表述中正确的是()。

A C++已有的任何运算符都可以重载 B 运算符函数的返回类型不能声明为基本数据类型
C 在类型转换函数的定义中不需要声明返回类型 D 可以通过运算符重载来创建C++中原来没有的运算符 参考答案 C

有如下程序：

```
#include<iostream>

using namespace std;

class Complex
```

```

{
    double re, im;
public:
    Complex(double r, double i):re(r), im(i) { }
    double real() const { return re; }
    double image() const { return im; }
    Complex& operator+=( Complex a )
    {
        re+=a.re;
        im+=a.im;
        return *this; }
};

ostream& operator<<(ostream& s, const Complex& z)
{
    return s<<' ('<<z.real()<<', '<<z.image()<<')' ; }

int main( )
{
    Complex x(1, 2), y(2, 3);
    cout<<(x+=y)<<endl;
    return 0;}

```

执行这个程序的输出结果是()。

A (1, -2) B (2, 3) C (3, 5) D (3, 1) 参考答案 C

下列函数中，能重载运算符的函数是_____。

A 虚函数 B 构造函数 C 友元函数 D 析构造函数 参考答案 C

下列关于运算符重载的描述中，正确的是_____。

A 运算符重载可以改变操作数个数 B 运算符重载可以改变优先级 C 运

算符重载可以改变结合性 D 运算符重载不可以改变语法结构 参考答案 D

在下列运算符中，不能重载的是_____。

A <= B >> C :: D &= 参考答案 C

下列运算符中全都可以被友元函数重载的是()。

A =, +, -, \ B [], +, (), new C ->, +, *, >> D <<, >>, +, * 参考答案 D

假定要对类AB定义加号操作符重载成员函数，实现两个AB类对象的加法，并返回相加结果，则该成员函数的声明语句为()。

A AB operator+(AB &a, AB &) B AB operator+(AB &) C opetacor+(AB a) D AB &operator+() 参考答案 B

一个函数为 void f(int, char ch='a'), 另一个函数为 void f(int), 则它们()。

A 不能在同一程序中定义 B 可以在同一程序中定义并可重载 C 可以在同一程序中定义，但不可以重载 D 以上三种说法均不正确 参考答案 B

下列是重载乘法运算符的函数原型声明，其中错误的是()

。 A MyClass operator*(double, double) ; B MyClass operator*(double, MyClass); C MyClass operator*(MyClass,double); D MyClass operator*(MyClass, MyClass); 参考答案 A

() 既可重载为一元运算符，又可重载为二元运算符。 A * B = C = 和 * D * 和 ++ 参考答案 A

下面有关重载的说法中，错误的是()。

A 函数重载要求同名函数在参数个数或参数类型上不同 B

运算符重载是用同一个运算符针对不同类型数据进行不同的运算操作

C 所有的运算符都可以重载D 运算符重载函数通常是类的成员函数和友元函数 参考答案 C

重载函数在调用时选择的依据中，错误的是()。

A 参数个数 B 参数的类型 C 函数名字 D 函数的类型 参考答案 D

成员函数重载的定义与调用和普通函数重载的定义与调用是（ ）。

A 相似的 B 不一样的 C 完全相同的 D 以上都不正确 参考答案 A

下面对C++运算符的描述正确的是()

A 只有类成员运算符。 B 只有友元运算符。 C 只有非成员和非友元运算符。 D 上述三项都有。 参考答案 D

下面对运算符重载的描述正确的是（ ）

A 运算符重载可以改变优先级。 B 运算符重载可以改变结合性。 C 运算符重载可以改变语法结构。 D 运算符重载不可以改变运算符的操作数的个数。 参考答案 D

如下程序定义了“单词”类Word，类中重载了<运算符，用于比较“单词”的大小，返回相应的逻辑值。程序的输出结果为：After

Sorting: Happy Welcome，请将程序补充完整。

```
# include <iostream>

# include <string>

using namespace std;

class Word{

public:

Word(string s) : str(s) {}

string getStr() { return str; }

    __1__

const {return(str<w.str);}

friend ostream& operator<<(ostream& output, const Word &w)

{ output<<w.str; return output; }

private:
```

```

string str;
};

int main() {
    Word w1("Happy"), w2("Welcome");
    cout<<"After sorting: ";
    if(w1<w2) cout<<w1<<' ' <<w2;
    else cout<<w2<<' ' <<w1;
    return 0;
}

```

参考答案1: bool operator < (Word w);

“图形”类Shape中定义了纯虚函数CalArea(), “三角形”类Triangle继承了类Shape, 请将Triangle类中的CalArea函数补充完整。

```

class Shape{
public:
    virtual int CalArea()=0;
};

class Triangle: public Shape{
public:
    Triangle(int s, int h) : side(s), height(h) {}

    __1__
    { return side * height/2;}

private:
    int side;
}

```

```
int height;  
};
```

参考答案1: int CalArea(size,height)

双目运算符重载函数为成员函数时，重载函数有__1__个参数。

参考答案1: 1 或 壹 或 一

运算符重载函数通常为类的__1__函数和友元函数

。参考答案1: 成员

运算符重载函数调用表达式operator+(x,y)可以表示为__1__。

参考答案1: x+y

重载的关系运算符和逻辑运算符的返回类型应当是__1__。

参考答案1: bool类型或bool型或布尔或布尔类型或布尔型 或 逻辑或逻辑型 或 逻辑类型 或 bool

__1__函数重载将会给初始化带来多种方式。参考答案1: 构造

运算符可重载类运算符和__1__运算符。参考答案1: 友元

重载的运算符保持其原有的__1__、优先级和结合性不变。

参考答案1: 操作数个数

运算符[]只能用__1__运算符重载。参考答案1: 类

etegdfgdgdfgdgdfgsjdfdsjfkds 参考答案 sdfdf

下列关于模板形参的描述中，错误的是()。

A 模板形参表必须在关键字template之后 B 模板形参表必须用括弧()括起来 C 可以用class修饰模板形参 D 可以用typename修饰模板形参

参考答案 B

下列关于函数模板的描述中，错误的是

A 从模板实参表和从模板函数实参表获得信息矛盾时，以模板实参的信

息为准 B 对于常规参数所对应的模板实参，任何情况下都不能省略
C 虚拟类型参数没有出现在模板函数的形参表中时，不能省略模板实参
D 模板参数表不能为空 参考答案 A

有如下函数模板声明：

```
template<typename T>  
T Max(T a , T b)  
{ return (a>=b) ? a : b; }
```

下列对函数模板Max()的调用中错误的是()。 A Max(3.5 , 4.5) B Max(3.5 , 4) C Max<double>(3.5 , 4.5) D Max<double>(3.5 , 4) 参考答案 B

()是以一种完全通用的方法来设计函数或类而不必预先说明将被使用的每个对象的类型。

A 模板 B 类 C 对象 D 函数 参考答案 A

有如下函数模板定义：

```
template<class T1, int a2, int a3>  
T1 fun(T1 a1)  
{  
    return (a1+a2+a3);  
}
```

则以下调用中正确的是()。

A fun<int, 4, 3>(5); B fun<4 , 3>(5); C fun<int, int, int>(5); D fun(5); 参考答案 A

关于在调用模板函数时模板实参的使用，下列表述中正确的是()。

A 对于虚拟类型参数所对应的模板实参，如果能从模板函数的实参中获得相同的信息，则都可以省略 B 对于虚拟类型参数所对应的模板实

参，如果它们是参数表中的最后的若干个参数，则都可以省略 C 对于虚拟类型参数所对应的模板实参，若能够省略则必须省略 D 对于常规参数所对应的模板实参，任何情况下都不能省略 参考答案 D

有如下函数模板定义：

```
template<class T>
T func(T x,T y)
{ return x*x+y*y; }
```

在下列对func的调用中，错误的是()。

A func(3, 5); B func(3.0, 5.5); C func(3, 5.5); D func<int>(3, 5.5); 参考答案 C

类模板的使用实际上是将类模板实例化成一个具体的()。

A 类 B 对象 C 函数 D 模板 参考答案 A

类模板的模板参数可用做_____。

A 数据成员的类型 B 成员函数的类型 C 成员函数的参数类型 D 以上均可 参考答案 D

下面关于类模板的派生与继承，描述错误的是_____。

A 类模板可以继承，继承的方法与普通的类一样 B 声明模板继承之前，必须重新声明类模板 C 模板类的基类和派生类不可以是非模板类 D 模板类的基类和派生类可以是模板类或非模板类 参考答案 C

下列说法中，不正确的是_____。

A 类模板可以从普通类派生，也可以从类模板派生 B 一个普通基类不能派生类模板 C 根据建立对象时的实际数据类型，编译器把类模板实例化为类模板 D 可以从构造函数参数列表推断出模板实例化参数类型 参考答案 D

下面关于类模板叙述不正确的是_____。

A 可以使用派生 B 可以有多个析构函数 C 可以有多个构造函数 D 可以有多个参数 参考答案 B

需要一种逻辑功能一样的函数，而编制这些函数的程序文本完全一样，区别只是数据类型不同。对于这种函数，下面不能 用来实现这一功能的选项是_____。

A 宏定义（宏函数） B 为各种类型都重载这一函数 C 模板 D 友元函数 **参考答案 D**

下面关于模板叙述不正确的选项是_____。

A 不可以通过一个类来派生类模板 B 可以从一个类模板派生出一个新的类模板 C 类模板支持单一继承 D 类模板可以有多个参数 **参考答案 A**

关于关键字class和typename，下列描述中正确的是()。

A 程序中所有的class都可以替换为typename B 程序中所有的typename都可以替换为class C 选项A和B都正确 D 选项A和B都错误 **参考答案 B**

关于关键字class和typename，下列表述中正确的是()。

A 程序中的typename都可以替换为class B 程序中的class都可以替换为typename C 在模板形参表中只能用typename来声明参数的类型 D 在模板形参表中只能用class或typename来声明参数的类型 **参考答案 A**

类模板的模板参数可用做()。

A 数据成员的类型 B 成员函数的类型 C 成员函数的参数 D 以上均可 **参考答案 D**

C++中模板可以分为()

A 类模板和函数模板 B 模板类和模板函数 C 类和函数 D 对象和函数 **参考答案 A**

() 使一个函数可以定义成对许多不同数据类型完成同一个任务。

A 函数模板 B 重载函数 C 递归函数 D 模板函数 **参考答案 A**

请将下列模板类Data补充完整。

```
template <typename T>
```

```
class Data{
```

```
public:
void put(T v) { val=v; }
__1__get() //返回数据成员val的值，返回类型不加转换
{ return val; }
private:
T val;
};
```

参考答案1: T

C++的模板机制包含两种类型，一种是__1__模板，另一种是类模板。 参考答案1: 函数模板 或 函数

以下函数模板max()的功能是返回数组a中最大元素的值。请将横线处缺失部分补充完整。

```
template<typename T>
T max(T a[ ], int n)
{
    T m=a[0];
    for(int i=1; i<n; i++)
        if(a[i]>m) __1__;
    return m;
}
```

参考答案1: m=a[i]

模板分为__1__模板和类模板。 参考答案1: 函数

模板声明中的第一个符号总是关键字__1__。 参考答案1: template
在模板声明中用"<"和">"括起来的部分是模板的__1__。

参考答案1：形参表 或 形参列表

有如下函数模板声明：

```
template<typename T>
T Max(T a,T b)
{return(a>=b)?a:b;}
```

函数模板的实例Max('A', 'F')的参数T的类型为__1__。(注意：用英文答题)

参考答案1：char

__1__是以一种完全通用的方法来设计函数或类而不必预先说明将被使用的每个对象的类型。参考答案1：模板

一个__1__允许用户为类定义一种模式，使得类中某些数据成员、成员函数参数和参数类型能取任意数据类型。参考答案1：类模板

模板分为函数模板和__1__模板。参考答案1：类

模板的形参可以分为虚拟类型参数和__1__。参考答案1：常规参数

阅读程序，严格按输出顺序与格式写出程序运行结果

```
#include <iostream.h>
template <class t,class z>
void Sj(t a,z b)
{ if (sizeof a>=sizeof b)
{a=a*b;cout<<"a="<<a;}
else
{b=a*b;cout<<"b="<<b;}
}
void main()
```

```
{ int x=2;
    double y=2.1;
    Sj(x,y);
} 参考答案 b=4.2
```

有如下程序：

```
# include <iostream>
using namespace std;
class AA
{
public:
    AA( ) { cout<<'1'; }
};
class BB: public AA
{
    int k;
public:
    BB( ) : k(0) { cout<<'2'; }
    BB(int n) : k(n) { cout<<'3'; }
};
int main( )
{
    BB b(4), c;
    return 0;
}
```

```
}
```

运行时的输出结果是()。

A 1312 B 132 C 32 D 1412 参考答案 A

有如下程序：

```
# include <iostream>

using namespace std;

class C1
{
public:
    ~C1( ) { cout<<1; }
};

class C2: public C1
{
public:
    ~C2( ) { cout<<2; }
};

int main( )
{
    C2 cb2;
    C1 *cb1;
    return 0;
}
```

运行时的输出结果是()。

A 121 B 21 C 211 D 12 参考答案 B

本题序号：67 通过派生类的对象可直接访问其

A 公有继承基类的公有成员 B 公有继承基类的私有成员 C 私有继承基类的公有成员 D 私有继承基类的私有成员 参考答案 A

本题序号：117 以下对派生类叙述不正确的是_____。

A 一个派生类可以作另一个派生类的基类 B 一个派生类可以有多个基类 C 具有继承关系时，基类成员在派生类中的访问权限不变 D 派生类的构造函数与基类的构造函数有成员初始化参数传递关系 参考答案 C

派生类的成员函数不能访问基类的_____。

A 公有成员和保护成员 B 公有成员 C 私有成员 D 保护成员 参考答案 C

下列描述中哪个是正确的_____。

A 私有派生的子类无法访问父类的成员 B 类A的私有派生子类的派生类C无法初始化其祖先类A对象的属性，因为类A的成员对类C是不可访问的 C 私有派生类不能作为基类派生子类 D 私有派生类的所有子孙类将无法继续继承该类的成员 参考答案 B

以下关于私有和保护成员的叙述中，不正确的是

A 私有成员不能被外界引用，保护成员可以 B 私有成员不能被派生类引用，保护成员在公有继承下可以 C 私有成员不能被派生类引用，保护成员在保护继承下可以 D 私有成员不能被派生类引用，保护成员在私有继承下可以 参考答案 A

以下关于派生机制的描述中，不正确的是

A 派生类不仅可以继承基类的成员，也可以添加自己的成员 B 设置protected成员是为派生类访问基类成员之用 C 采用不同的继承方式，将限制派生类对基类成员的访问 D 采用私有继承，派生类只能得到基类的公有成员 参考答案 D

若要用派生类的对象访问基类的保护成员，以下观点正确的是 A 不可

能实现 B 可采用保护继承 C 可采用私有继承 D 可采用公有继承 **参考答案 A**

下面有关派生类的说法中，错误的是_____。 A 一个派生类可以作为另一个派生类的基类 B 一个派生类可以有多个基类 C 派生类构造函数可以将初始化参数值传递给基类的构造函数和对象成员的构造函数 D 具有继承关系时，基类成员在派生类中的访问权限不变 **参考答案 D**

下面有关基类和派生类的说法中，错误的是_____。

A 派生类可以增添一些基类中没有的成员 B 派生类中可出现与基类同名的成员（若是成员函数则参数的个数和类型相同），但会出现同名覆盖现象；若要引用或调用基类中同名的成员，必须用访问域控制符（基类：：成员）加以指明 C 派生类可以重载基类中的成员函数 D 派生类可以有选择性地继承基类中的某些成员，删除不需要的成员 **参考答案 D**

派生类中的成员不能直接访问基类中的()成员。 A public B private C virtual D protected **参考答案 B**

当派生类从基类公有继承时，以下哪种说法是错误的。

A 可以用派生类对象为基类对象赋值。 B 可以用基类对象为派生类对象赋值。 C 可以把指向派生类对象的指针赋给基类对象的指针。 D 可以把派生类对象的地址赋给基类对象的指针。 **参考答案 B**

下列叙述中，错误的是()。

A 派生类可以使用private派生 B 对基类成员的访问必须是无二义性的 C 基类成员的访问能力在派生类中维持不变 D 派生类继承基类的所有成员 **参考答案 A**

派生类的构造函数的成员初始化列表中，不能包含()。 A 基类的构造函数 B 派生类中对象成员的初始化 C 基类中对象成员的初始化 D 派生类中一般数据成员的初始化 **参考答案 C**

下面关于多重继承的叙述，正确的是

A c++不支持多重继承。 B 多重继承是指从多层基类中派生出的子类。 C 多重继承中派生类成员的访问控制规则与单一继承规则不相同。

D 多重继承中可以通过虚基类防止二义性问题。 参考答案 D

下列描述中错误的是()。

A 继承分为多重继承和单继承 B 对象间通信靠传递消息实现 C 在外面看不到对象的内部特征是基于对象的“模块独立性好”这个特征 D 类是具有共同属性、共同方法的对象的集合 参考答案 C

派生类由public方式继承基类成员，则

A 基类中的私有成员可以被派生类的成员函数访问。 B 基类中的保护成员不能被派生类的成员函数访问。 C 基类中的公有成员可以被派生类的成员函数访问。 D 以上都不正确 参考答案 C

派生类由protected方式继承基类成员，则

A 基类中的私有成员成为派生类中的私有成员。 B 基类中的保护成员成为派生类中的私有成员。 C 基类中的公有成员成为派生类中的私有成员。 D 基类中的公有成员成为派生类中的保护成员。 参考答案 D

下列说法中错误的是()。

A 公有继承时基类中的public成员在派生类中仍是public成员 B 私有继承时基类中的protected成员在派生类中仍是protected成员 C 私有继承时基类中的public成员在派生类中仍是private成员 D 保护继承时基类中的public成员在派生类中仍是protected成员 参考答案 B

有以下定义的子类：

```
class A:public B
```

```
{ ... };
```

则下面描述中正确的是

A A类中的成员函数可以访问B类中的private 数据成员 B A类中的成员函数只能访问B类中的protected数据成员。 C A类中的成员函数可以访问B类中的public数据成员。 D A类中的成员函数只能访问B类中的public数据成员 参考答案 C

继承与派生可以给软件开发带来很多好处，下述中不正确的是： A 避免公用代码的重复开发，提高了程序开发的效率和质量。 B 可以减少

代码和数据的冗余。 C 通过层次关系组织对象，给编码与代码重用带来了方便。 D 软件的执行效率大大提高 参考答案 D

如果类之间具有继承和派生关系，那么它们之间应具有以下几个特征：

- 1) 类之间应具有共享的特征；
- 2) 类之间应具有差别或新增特征；
- 3) 类之间是一种层次结构；

上述中正确的是

A1)、2) B2)、3) C 1)、3) D 1)、2)、3) 参考答案 D

下面哪些对象属于继承与派生的关系：1) 交通工具与汽车 2) 汽车与轮胎 3) 动物与老虎 4) 电脑与内存

A1) 和2) B 3) 和4) C 1) 和3) D 2) 和4) 参考答案 C

设有基类定义：

```
class cBase{  
private: int a;  
protected: int b;  
public: int c;  
};
```

派生类采用何种继承方式可以使成员变量b成为自己的私有成员。

A私有继承 B 保护继承 C 公有继承 D 以上都可以 参考答案 A

有以下定义类及变量：

```
class B{  
public:  
B() {b=0;};  
protected::
```

```
int b;};  
class A:private B  
{  
public:  
int get() {return b;}  
};  
B cB;  
A cA;  
int a;
```

则下面语句中正确的是

A a=cA.b; B a=cA.get(); C a=cB.b; D以上都不对 **参考答案 B**

设有如下定义的基类与派生类:

```
class cBase{  
public: int x;  
protected: int y;  
private: int z; };  
class cDerived:protected cBase{  
public: void sety(int a) {y=a;};  
};
```

实例化对象:

```
cDerived cc;
```

则下面哪条语句是正确的:

A cc.x=1; B cc.y=2; C cc.z=3; D cc.sety(4); **参考答案 D**

下面关于基类与派生类的访问控制规则的叙述，错误的是

A 不管是何种派生，基类的private成员都不能被其派生类访问。 B 在派生类中访问基类的private成员可以通过基类的public和protected成员函数访问。 C 当通过private继承时，基类中的protected成员数据不能被派生类的成员函数访问。 D 当通过public继承时，基类中的protected成员数据可以被派生类的成员函数访问。 参考答案 C

下面叙述不正确的是_____。

A 派生类可以有自己的构造函数 B 派生类继承基类的构造函数 C 派生类继承基类的析构函数 D 派生类只能继承基类的一个构造函数 参考答案 C

下面叙述不正确的是_____。

A成员的访问能力在私有派生类中和公有派生类中是不同的 B基类的私有成员在公有派生类中不可访问 C赋值兼容规则不是用于多重继承的组合 D基类公有成员在保护派生中是保护的 参考答案 C

下面叙述不正确的是_____。

A 基类的保护成员在派生类中仍然是保护的 B 基类的保护成员在公有派生中仍然是保护的 C 基类的保护成员在私有派生中是私有的 D 对基类成员的访问必须是无二义性的 参考答案 A

下列对派生类的描述中，错误的是_____。

A一个派生类可作为另一个派生类的基类 B派生类至少有一个基类 C派生类成员除了它自己的成员外，还包含它的基类成员 D派生类中继承的基类成员的访问权限到派生类保持不变 参考答案 D

派生类的构造函数的初始化列表中，不能含_____。

A 基类的构造函数 B 派生类中子对象的初始化 C 基类的子对象初始化 D 派生类中一般数据成员的初始化 参考答案 C

有如下程序：

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

```
class TestClass
```

```

{
private:
    char c;public:
    TestClass (char n):c(n) { }
    ~TestClass ( ) { cout<<c; }
};

class TestClass1:public TestClass
{
private:
    char c;
public:
    TestClass1(char n):TestClass (n+1), c(n) { }
    ~TestClass1( ) { cout<<c; }
};

void main( )
{ TestClass1 obj( 'x' ); }

```

执行上面的程序将输出()。

A xy B yx C x D y **参考答案** A

在多重继承中，公有派生和私有派生对于基类成员在派生类中的访问性与单一继承的规则_____。

A 相同 B不同 C部分相同，部分不同 D以上都不对 **参考答案** A

下面叙述不正确的是_____。

A 派生类一般都用公有派生 B 对基类成员的访问必须是无二义性的
C 赋值兼容规则也适用于多重继承的组合 D 基类的公有成员在派生类

中仍然是公有的 参考答案 D

有如下程序：

```
#include<iostream.h>

class A
{
public:
    virtual void func1( ) {cout<<"A1";}
    void func2( ) {cout<<"A2";}
};

class B:public A
{
public:
    void func1( ) {cout<<"B1";}
    void func2( ) {cout<<"B2";}
};

void main( )
{
    A *p=new B;
    p->func1( );
    p->func2( );
}
```

执行该程序，屏幕上将显示输出()。 A B1B2 B A1A2 C B1A2
D A1B2 参考答案 C

能被派生类继承的函数是_____。

A 静态成员函数 B 构造函数 C 析构函数 D 复制的构造函数 参考答案
A

不论派生类以何种方式继承，都不能使用基类的_____。

A 公有成员 B 私有成员 C 保护成员 D 公有成员和保护成员 参考答案
B

下面关于继承的描述中，错误的是_____。

A 继承描述的是类的层次关系，派生类可以具有与基类相同的属性和方法
B 一个基类可以有多个派生类，一个派生类可以有多个基类
C C++有两种继承包括单一继承和多重继承 D 静态成员不可以被继承
参考答案 D

在C++中，类与类之间的继承关系具有_____。

A 自反性 B 对称性 C 传递性 D 规律性 参考答案 C

以下叙述错误的是_____。

A 基类的保护成员在公有派生类中仍然是保护成员 B 基类的保护成员在派生类中仍然是保护成员
C 基类的保护成员在私有派生中是私有成员 D 基类的保护成员不能被派生类的对象访问 参考答案 B

有如下程序：

```
#include <iostream.h>

class Base
{
public:
    Base( )
    { cout<<"BB";
      f( ); }
    void f( ){ cout <<"Bf"; }
```

```
};

class Derived: public Base
{
public:
    Derived( ) { cout<<"DD"; }
    void f( ) { cout<<"Df"; }
};

void main( )
{ Derived d; }
```

执行上面的程序将输出()。

A BBBfDD B BBDfDDDf C DD D DDBBBf 参考答案 A

下列描述中()是正确的。

A 私有派生的子类无法访问父类的成员 B 类A的私有派生子类的派生类C无法初始化其祖先类A对象的属性，因为类A的成员对类C是不可访问的
C 私有派生类不能作为基类派生子类D 私有派生类的所有子孙类将无法继续继承该类的成员 参考答案 B

派生类的成员函数不能访问基类的()。

A 公有成员和保护成员B 公有成员 C 保护成员D 私有成员 参考答案 D

派生类的对象对它的基类成员中是可以访问的()。

A 公有继承的公有成员B 公有继承的私有成员 C 公有继承的保护成员
D 私有继承的公有成员 参考答案 A

以下对派生类叙述不正确的是()。

A 一个派生类可以作另一个派生类的基类B 一个派生类可以有多个基类
C 具有继承关系时，基类成员在派生类中的访问权限不变D 派生类的构造函数与基类的构造函数有成员初始化参数传递关系

参考答案 C

以下关于派生机制的描述中，不正确的是（ ）。

A 派生类不仅可以继承基类的成员，也可以添加自己的成员
B 设置protected成员是为派生类访问基类成员之
C 采用不同的继承方式，将限制派生类对基类成员的访问
D 采用私有继承，派生类只能得到基类的公有成员
参考答案 D

C++语言建立类族是通过（ ）实现的。

A 类的嵌套 B 虚函数 C 类的继承 D 抽象类 参考答案 C

下面关于继承的描述中错误的是（ ）

A 析构函数不能被继承。 B 派生类是基类的组合。 C 派生类的成员除了它自己的成员外，还包含了它的基类成员。 D 派生类中继承的基类成员的访问权限到派生类保持不变。 参考答案 D

下面对于析构函数的描述错误的是（ ）。

A 类中有且仅有一个析构函数。 B 析构函数可以有形参。 C 析构函数没有函数类型。 D 析构函数在对象消失时被自动执行。 参考答案 B

下列对派生类的描述中，错误的是（ ）。

A 一个派生类可以作为另一个派生类的基类。 B 派生类至少有一个基类。 C 派生类的缺省继承方式是private。 D 派生类只继承了基类的公有成员和保护成员。 参考答案 D

继承的方式有公有继承、私有继承和__1__3种。

参考答案1：保护继承

派生类的成员一般分为两部分，一部分是从基类__1__的成员，另一部分是自己定义的新成员。 参考答案1：派生 或 继承

由保护派生得到的派生类，它从基类继承的公有和保护成员都将变为派生类的__1__。 参考答案1：保护成员

生成一个派生类对象时，先调用__1__的构造函数，然后调用派生类的构造函数。 参考答案1：基类

在公有继承关系下，派生类的对象可以访问基类中的__1__成员。

参考答案1：公有

对于派生类的构造函数，在定义对象时构造函数的执行顺序为：先执行调用__1__的构造函数，再执行调用对象成员的构造函数，最后执行派生类的构造函数体中的内容。参考答案1：基类

如果希望从基类继承的公有成员，只能被派生类及派生类的子类中的所有非静态成员函数和友元函数直接访问，其它的外部函数不能访问，则应使用__1__继承。参考答案1：protected 或保护

在声明派生类时，如果不显式地给出继承方式，缺省的类继承方式是私有继承private。已知有如下类定义：

```
class Base
{
    protected:
        void fun( ) {}
};

class Derived : Base
{
};
```

则Base类中的成员函数fun()，在Derived类中的访问权限是__1__（注意：要求填写private、protected或public中的一项）。

参考答案1：private

请将下列类定义补充完整。

```
class Base
{
    public:
        void fun( )
```

```

        {
            cout <<"Base::fun"<<endl;
        }
};

class Derived:public Base
{
public:
    void fun( )
    {
        __1__ ;    //显式调用基类的fun函数
        cout<<"Derived::fun"<<endl;
    }
};

```

参考答案1: Base::fun()

在多继承中基类和派生类中同时出现__1__存在二义性。

参考答案1: 同名函数

虚函数必须是类的__1__。参考答案1: 非静态成员函数或成员函数

从多个基类中派生出新的子类，这种派生方法称为__1__。

参考答案1: 多重继承

派生类是基类的组合，多继承可以看作是多个__1__的简单组合。

参考答案1: 单继承

虚函数不得声明为__1__函数。参考答案1: 静态

构造函数不能被继承，析构函数__1__被继承。参考答案1: 不能

公有派生类的对象可以初始化__1__。 **参考答案1**：基类的引用
引进虚基类后，派生类的对象中存在__1__个虚基类的子对象。

参考答案1：一

阅读程序，写出运行后的输出结果

```
#include<iostream>

using namespace std;

class A{
public:
    ~A() {cout<<"destructing A"<<endl;}
};

class B:public A{
public:
    ~B() {cout<<"destructing B"<<endl;}
};

int main() {
    B b1;
    return 0;
}
```

参考答案 destructing B destructing A

阅读程序，写出运行后输出的结果：

```
#include<iostream>

using namespace std;

class A{
```

```

public:
    A(int i) {a=i;b=0;}
    void display( ) {cout<<"a"<<a<<"b"<<b;}
private:
    int a; int b;
};

class B:public A{
public:
    B(int i,int j):A(i) {c=j;}
    void display1( ) {display();  cout<<"c"<<c<<endl; }
private:
    int c;
};

int main() {
    B b1(1,2);
    b1.display1( );
    return 0;
}  参考答案 a=1b=0c=2

```

阅读下面的程序，写出运行后输出的结果：

```

#include<iostream>

using namespace std;

class A{
public:

```

```

        A() {cout<<"constructing A"<<endl;}
};
class B:public A{
public:
        B() {cout<<"constructing B"<<endl;}
};
int main() {
B b1;
return 0;
}

```

参考答案 constructing A constructing B

阅读程序，写出运行后输出的结果：

```

#include<iostream>
using namespace std;
class A{
public:
A() {a=0;b=0;}
        void display( ) {cout<<"a="<<a<<"b="<<b;}
private:
        int a; int b;
};
class B:public A{
public: B( ) {c=0;}

```

```

void display1( ) {display( );  cout<<"c"<<c<<endl; }
private:
int c;};
int main( ){
    B b1;
    b1.display1( );
    return 0;
}  参考答案 a=0b=0c=0

```

阅读程序，写出运行后输出的结果：

```

#include<iostream>
using namespace std;
class A{
public:
    A(int i) {a=i;b=0;}
    void display( ) {cout<<"a"<<a<<"b"<<b;}
private:
    int a; int b;
};
class B:public A{
public:
    B(int i):A(i){c=0;}
    void display1( ) {
        display( );
    }
};

```

```
cout<<"c="<<c<<endl;
}
```

```
private:
```

```
    int c;
};
```

```
int main( ){
```

```
    B b2(1);
```

```
    b2.display1( );
```

```
    return 0;
```

```
} 参考答案 a=1b=0c=0
```

有如下程序：

```
#include<iostream>
```

```
using namespace std;
```

```
class TestClass
```

```
{
```

```
protected:
```

```
    TestClass ( ) {cout<<'x' ;}
```

```
    TestClass (char c) {cout<<c;}
```

```
};
```

```
class TestClass1:public TestClass
```

```
{
```

```
public:
```

```
    TestClass1(char c) {cout<<c;}
```



```
};

void main( )
{
    TestClass1 d1('y');
}
```

执行这个程序，屏幕上将显示输出是 参考答案 xy
有如下程序：

```
#include<iostream.h>

class TestClass
{
    int a;
public:
    TestClass (int x){ a=x; }
    void show( ){ cout<<a; }
};

class TestClass1:public TestClass
{
    int b;
public:
    TestClass1(int i) : TestClass (i+1),b(i){ }
    void show( ){cout<<b;}
};

void main( )
```

```

{
    TestClass b(5), *p;
    TestClass1 d(1);
    p=&d;
    p->show( );
}

```

该程序运行后的打印结果是。 参考答案 2

有如下程序：

```

#include <iostream.h>

class ONE
{
public:
    virtual void f( ) { cout << "1"; }
};

class TWO : public ONE
{
public:
    TWO( ) { cout << "2"; }
};

class THREE : public TWO
{
public:
    virtual void f( )

```

```

        { TWO::f( );
            cout << "3"; }
};

void main( )
{
    ONE aa, *p;
    TWO bb;
    THREE cc;
    p = &cc;
    p->f( );
}

```

写出执行该程序的输出结果。 参考答案 2213

有如下程序：

```

#include<iostream.h>

class TestClass1
{
public:
    TestClass1( ) {cout<<"A";}
};

class TestClass2
{
public:
    TestClass2( ) { cout<<"B"; }
}

```

```
};
class TestClass3:public TestClass1
{
    TestClass2 b;
public:
    TestClass3( ) { cout<<"C"; }
};
void main( )
{
    TestClass3 obj;
}
```

写出该程序的执行后的输出结果。 参考答案 ABC

下面4个选项中，（ ）是用来声明虚函数的。 A virtual
B public C using D false 参考答案 A

下列关于动态联编的描述中，错误的是（ ）

A 动态联编是以虚函数为基础 B 动态联编是运行时确定所调用的函数代码的 C 动态联编调用函数操作是指向对象的指针或对象引用 D 动态联编是在编译时确定操作函数的 参考答案 D

如果一个类至少有一个纯虚函数，那么就称该类为（ ）。 A 抽象类 B 虚函数 C 派生类 D 以上都不对 参考答案 A

C++类体系中，不能被派生类继承的有（ ）。

A构造函数 B虚函数 C静态成员函数 D赋值操作函数 参考答案 A

下列虚基类的声明中，正确的是（ ）。

A: class virtual B: public A B: virtual class B: public AC class
B: public A virtual D class B: virtual public A 参考答案 D

关于虚函数的描述中，（ ）是正确的。

A 虚函数是一个静态成员函数 B 虚函数是一个非成员函数 C 虚函数既可以在函数说明时定义，也可以在函数实现时定义 D 派生类的虚函数与基类中对应的虚函数具有相同的参数个数和类型 参考答案 D

有关公用体与结构体不正确的说法是_____。

A 都可有多个成员 B 结构体可同一时刻保留多个数据 C 公用体同一时刻只能保留一个数据 D 结构体各成员的地址相同 参考答案 D

下列特征中不是面向对象方法的主要特征的是（ ）。 A 多态性 B 继承 C 封装性 D 模块化 参考答案 D

下列描述中，属于抽象类的性质是_____。

A 可以说明虚函数 B 可以进行构造函数重载 C 可以定义友元函数 D 不能说明其对象 参考答案 D

要实现动态联编必须_____。

A 通过成员名限定来调用虚函数 B 通过对象名来调用虚函数 C 通过派生类对象来调用虚函数 D 通过对象指针或引用来调用虚函数 参考答案 D

以下哪个基类中的成员函数表示纯虚函数_____。 A `virtual void tt()=0;` B `void tt(int)=0;` C `virtual void tt(int);` D `virtual void tt(int){};` 参考答案 A

如果一个类中至少有一个纯虚函数，那么就称该类为_____。 A 抽象类 B 虚基类 C 派生类 D 以上都不对 参考答案 A

关于动态联编的下列叙述中，错误的是_____。

A 动态联编是以虚函数为基础的 B 动态联编运行时确定所调用函数代码 C 动态联编调用函数操作是指向对象的指针或对象引用 D 动态联编是在编译时确定操作函数 参考答案 D

在类中用来声明虚成员函数的关键字是_____。

A `virtual` B `public` C `protected` D `private` 参考答案 A

下面正确的说法是_____。

A 关键字virtual只能用来声明虚函数 B 声明虚函数时，才使用关键字virtual C 关键字virtual不能用来声明虚基类 D 关键字virtual能用来声明虚基类 **参考答案 D**

有如下程序：

```
#include<iostream>

using namespace std;

class TestClass
{
public:
    virtual void fun1( ) { cout<<"fun1TestClass"; }
    virtual void fun2( ) { cout<<"fun2TestClass"; }
};

class TestClass1:public TestClass
{
    void fun1( ) { cout<<"fun1TestClass1"; }
    void fun2(int x) { cout<<"fun2TestClass1"; }
};

int main( )
{
    TestClass obj1,*p;
    TestClass1 obj2;
    p=&obj2;
    p->fun1( );
```

```
p->fun2();  
return 0;  
}
```

该程序执行后的输出结果是()。

A fun1TestClass1 fun2TestClass B fun1TestClass1
fun2TestClass1 C fun1TestClass fun2TestClass D fun1TestClass
fun2TestClass1 参考答案 A

关于纯虚函数，下列表述中正确的是()。

A 纯函数是没有给出实现版本（即无函数体定义）的虚函数 B 纯
虚函数的声明总是以“=0”结束 C 派生类必须实现基类的纯虚函数 D 含
有纯虚函数的类不可能是派生类 参考答案 B

在派生类中定义虚函数时，可与基类中相应的虚函数不同的是
_____。

A 参数类型 B 参数个数 C 函数名称 D 函数体 参考答案 D

下列关于虚函数的描述中，正确的是()。

A 虚函数是一个static类型的成员函数 B 虚函数是一个非成员函数
C 基类中采用virtual说明一个虚函数后，派生类中定义相同原型的函
数时可不必加virtual说明 D 派生类中的虚函数与基类中相同原型的虚
函数具有不同的参数个数或类型 参考答案 C

下列关于虚基类的描述，错误的是()。

A 设置虚基类的目的是为了消除二义性 B 虚基类的构造函数在非虚基类
之后调用 C 若同一层中包含多个虚基类，这些虚基类的构造函数按它们
说明的次序调用 D 若虚基类由非虚基类派生而来，则仍然先调用基类构
造函数，再调用派生类的构造函数 参考答案 B

在类中用来声明虚成员函数的关键字是()。

A virtual B public C private D protected 参考答案 A

关于虚函数的描述中，()是正确的。

A 虚函数是一个静态成员函数B虚函数是一个非成员函数C虚函数既可以在函数说明时定义，也可以在函数实现时定义D派生类的虚函数与基类中对应的虚函数具有相同的参数个数和类型 **参考答案** D

下列虚基类的声明中，正确的是（ ）。

Aclass virtual B:public A Bvirtual class B:public A Cclass B:public A virtual Dclass B: virtual public A **参考答案** D

非成员函数应该声明为类的__1__才能访问该类的私有成员。

参考答案1: 友元函数

如果在类中定义了一个成员函数为__1__，则表明在该继承层次链条的派生类中有可能重新定义这个成员函数的实现，即它可 能被派生类的同名函数所覆盖。 **参考答案1**: 虚函数

虚函数从表现形式上看是指那些被__1__关键字修饰的成员函数。

参考答案1: virtual

执行下列程序：

```
#include<iostream>

using namespace std;

class TestClass1
{public:
    void fun1( ){ cout<<" TestClass1 "; }
    virtual void fun2( ) { cout<<" TestClass1 "; }
};

class TestClass2 : public TestClass1
{
public:
    void fun1( ){ cout<<" TestClass2 "; }
```



```

        void fun2( ) { cout<<" TestClass2  "; }
};
void f(TestClass1 & b)
{
    b.fun1( );
    b.fun2( );
}
int main( )
{
    TestClass2 obj;
    f(obj);
    return 0;
}

```

其输出结果是__1__。(注意：两项输出之间留一个空格)

参考答案1: TestClass1 TestClass2

动态多态性通过__1__实现。 **参考答案1:** 虚函数

如果不使用多态机制，那么通过基类的指针虽然可以指向派生类对象，但是只能访问从基类继承的成员。下列程序没有使用多态机制：

```

include <iostream>

using namespace std;

class Base
{

```

```

public:
    void print ( ) { cout << 'B'; }
};

class Derived : public Base
{
public:
    void print ( ) { cout << 'D'; }
};

int main( )
{
    Derived * pd = new Derived( );
    Base * pb = pd;
    pb->print( );
    pd->print( );
    delete pd;
    return 0;
}

```

其输出结果是__1__ 。

参考答案1: BD

定义一个函数名为fun，返回值为int，没有参数的纯虚函数的定义是__1__。

参考答案1: virtual int fun()=0**或**virtual int fun()=0;

如果类包含了一个或多个__1__，则它是抽象类。

参考答案1：纯虚函数

编译时的多态性可以通过使用__1__和类对象获得。

参考答案1：虚函数

已知类sample是一个抽象类，其成员函数display是无形参、无返回类型的纯虚函数，请完成其声明。

```
class sample
{
public:
    sample( ){ };
    __1__ ;

};
```

（注意：各项之间只用一个空格分隔）

参考答案1：virtual void display()=0

多态性分为两类：编译时的多态性和__1__的多态性。

参考答案1：运行时

在面向对象的程序设计中，不同的对象可以调用相同名称的函数并导致完全不同的行为现象称为__1__。参考答案1：多态性

成员函数有实函数、虚函数和__1__3种。参考答案1：纯虚函数

动态联编的多态性由__1__支持。参考答案1：虚函数

C++语言支持的两种多态性分别是__1__的多态性和运行时的多态性。参考答案1：编译时

如果一个类至少有一个纯虚函数，那么就称该类为__1__。

参考答案1：抽象类

虚函数必须是类的__1__ 参考答案1: 成员函数

含有纯虚函数的类称为__1__。 参考答案1: 抽象类

在C++中运行时的多态性是通过__1__来体现的。

参考答案1: 虚函数

在C++中, 编译时的多态性是通过函数重载和__1__体现的。

参考答案1: 模板

下列关于C++流的描述中, 错误的是

A cout>>'A' 表达式可输出字符A B eof()函数可以检测是否到达文件尾
C 对磁盘文件进行流操作时, 必须包含头文件fstream D 以
ios_base::out模式打开的文件不存在时, 将自动建立一个新文件 参考
答案 A

假定下列语句都是程序运行后首次执行的输出语句, 其中输出结果与另
外三条语句不同的语句是 A cout<<setfill('*')<<123<<setw(9)
<<321; B cout<<setfill('*')<<setw(6)<<left<<123<<setw(6)
<<right<<321; C cout<<123<<setfill('*')<<setw(6)<<321;
D cout<<setfill('*')<<setw(9)<<left<<123<<321; 参考答案 C

当使用ifstream 流类定义一个流对象并打开一个磁盘文件时, 文件的
默认打开方式为

A ios_base::in B ios_base::in | ios_base::out C ios_base::out
D ios_base::in & ios_base::out 参考答案 A

要建立文件流并打开当前目录下的文件file. dat 用于输入, 下列语句
中错误的是

A ifstream fin=ifstream.open ("file.dat"); B ifstream *fin=new
ifstream ("file.dat"); C ifstream fin; fin.open ("file.dat");
D ifstream *fin=new ifstream(); fin->open ("file.dat"); 参考答
案 D

有语句定义“char a[16];”, 如果从键盘输入一个含空格的字串, 可
用语句_____。

A `cin>>a;` B `cin>>a[0];` C `cin.get(a,16)` ;
D `cin.get(a[0],16);` 参考答案 C

当要用I/O流操作文件时，需要包含_____文件。 A `iso.h`
B `fstream.h` C `iostream.h` D `cmath` 参考答案 B

当要建立一个文件，并从其中读数据和向其中写数据时，应该建立下面哪种流类的对象_____。

A `ifstream` B `ofstream` C `fstream` D `ios` 参考答案 C

`cin` 是 I/O 流库预定义的_____。

A 类 B 对象 C 包含文件 D 常量 参考答案 B

以下关于文件操作的叙述中，不正确的是（ ）。

A 打开文件的目的是使文件对象与磁盘文件建立联系 B 文件读写过程中，程序将直接与磁盘文件进行数据交换 C 关闭文件的目的之一是保证将输出的数据写入硬盘文件 D 关闭文件的目的之一是释放内存中的文件对象 参考答案 B

若有语句

```
char str[20]; cin>>str;
```

当输入为：

This is a C++ program

时，`str`所得结果是（ ）

A This is a C++ program B This C This is D This is a C 参考答案 B

下列函数中，（ ）是对文件进行写操作的。

A `read()` B `seekg()` C `get()` D `put()` 参考答案 D

若磁盘上已存在某个文本文件，它的全路径文件名为
`d:\kaoshi\test.txt`，则下列语句中不能打开这个文件的是（ ）。

A ifstream file(“d:\kaoshi\test.txt”); B ifstream
file(“d:\\kaoshi\\test.txt”); C ifstream
file;file.open(“d:\\kaoshi\\test.txt”); D ifstream
*pFile=new ifstream(“d:\\kaoshi\\test.txt”); 参考答案 A

cout是I/O流库预定义的（ ）。

A 类 B 对象 C 包含文件 D 常量 参考答案 B

在下面格式化命令的解释中，错误的是（ ） A ios::skipws 跳过
输入中的空白字符 B ios::fill()获得当前的填充字符 C ios::hex转
换基数为八进制形式 D ios::precision设置当前的精度 参考答案 B

下列关于输入流类成员函数getline()的描述中，错误的是（
）。

A 该函数是用来读取键盘输入的字符串的。 B 该函数读取的字符串长
度是受限制的。 C 该函数读取字符串时，遇到终止符便停止。 D 该函
数读取字符串时，可以包含空格。 参考答案 A

语句ofstream f(“SALARY.DAT”,ios::app)的功能是建立流对象f，
并试图打开文件SALARY.DAT 与f 关联，而且（ ）。

A 若文件存在，将其置为空文件；若文件不存在，打开失败 B 若文件
存在，将文件指针定位于文件尾；若文件不存在，建立一个新文件
C 若文件存在，将文件指针定位于文件首；若文件不存在，打开失败
D 若文件存在，打开失败；若文件不存在，建立一个新文件 参考答案
B

进行文件操作时，需要包含（ ）文件。

A iostream.h B fstream.h C stdion.h D math.h 参考答案 B

关于read(char *buf,int size)函数的下列描述中，（ ）是对
的。

A 函数只能从键盘输入中获取字符串 B 函数所获取的字符多少是不受
限制的 C 该函数只能用于文本文件的操作中 D 该函数只能按规定读取
所指定的字符数 参考答案 D

当需要打开A盘上的xxk.dat文件用于输入时，则定义文件流对象的语句

为（ ）。

A fstream fin(“A:\\xxk.dat”); B ofstream
fin(“A:\\xxk.dat”); C ifstream fin(“A:\\xxk.dat” ,
ios::app); D ofstream fin(“A:\\xxk.dat” , ios::nocreate); 参
考答案 C

C++流中重载了运算符<<, 它是一个（ ）。 A 用于输出操作
的成员函数 B 用于输入操作的非成员函数 C 用于输入操作的成员函数
D 用于输出操作的非成员函数 参考答案 A

当使用ifstream流类定义一个流对象并打开一个磁盘文件时, 文件的隐
含打开方式为（ ）。

A ios::in B ios::out C ios::in | ios::out D 没有 参考答案 A

当使用fstream流类定义一个流对象并打开一个磁盘文件时, 文件的隐
含打开方式为（ ）。

A ios::in B ios::out C ios::in | ios::out D 没有 参考答案 C

下列输出字符d的方法中, （ ）是错误的。

A cout<<put('d'); B cout<<'d'; C cout.put('d'); D char a='d';
cout<<a; 参考答案 A

运行下列程序结果为（ ）。

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

```
int main( )
```

```
{
```

```
    cout.width(6);
```

```
    cout.fill('*');
```

```
    cout << 'a' << 1 << endl;
```

```
    return 0;
```

```
}
```

A *****a*****1 B*****a1 Ca*****1***** Da*****1 参考答案 D

在ios中提供控制的标志中，（ ）是转换为十六进制形成的标志位。

A hex B oct C dec D left 参考答案 A

1. 下面关于C++流的叙述中，正确的是（ ）

A cin是一个输入流对象。B可以用ifstream定义一个输出流对象。C执行语句序列char *y="PQMN"; cout<<y; D 执行语句序列char x[80]; cin.getline(x,80); 时，若键Happy new year 则x中的字符串是"Happy" 参考答案 A

使用setw()对数据进行格式输出时，应包含（ ）文件。

A iostream.h B fstream.h C iomanip.h D stdlib.h 参考答案 C

下列关于getline()函数的描述中，错误的是（ ）。A 该函数可以用来从键盘上读取字符串 B 该函数读取的字符串长度是受限制的 C 该函数读取字符串时，遇到终止符时便停止 D 该函数中所使用的终止符只能是换行符 参考答案 D

在进行了任何C++流的操作后，都可以用C++流的有关成员函数检测流的状态，其中只能用于检测刚进行的操作是否失败 的函数名是（ ）。

A fail B eof C bad D good 参考答案 A

下列关于C++流的说明中，正确的是（ ）。

A 与键盘、屏幕、打印机和通信端口的交互都可以通过流类来实现 B 从流中获取数据的操作称为插入操作，向流中添加数据的操作称为提取操作 C cin是一个预定义的输出流类 D 输出流有一个名为open的成员函数，其作用是生成一个新的流对象 参考答案 A

当从键盘输入yourname@thedomain.com 时，执行语句cin.getlin(ch,20,'@'); 后，字符数组 ch 的值是（ ）。

A yourname@domain B y C yournameD yourname@thedomain.c 参考答

案 C

C++流中重载了运算符<<，它是一个()。

A 用于输出操作的成员函数 B 用于输入操作的成员函数 C 用于输入操作的非成员函数 D 用于输出操作的非成员函数 参考答案 D

对于语句 `cout<<endl<<x;` 中的各个组成部分，下列叙述中错误的是()。

A `cout` 是一个输出流对象 B `endl` 的作用是输出回车换行 C `x` 是一个变量 D << 称作提取运算符 参考答案 D

执行语句序列

```
ofstream outfile("DATA.DAT");  
if(...) cout<<"OK"; else cout<<"FAIL";
```

后，如果文件打开成功显示“OK”，否则就显示“FAIL”。由此可知，上面if 语句的...处的表达式应是()。

A `outfile.fail()` 或 `outfile` B `outfile.good()` 或 `!outfile`
C `outfile.good()` 或 `outfile` D `outfile.fail()` 或 `!outfile` 参考答案 C

下列关于ios类的说法，正确的是()。

A 它是ostream 类和 istream 类的虚基类 B 它只是istream 类的虚基类 C 它只是ostream 类的虚基类 D 它是iostream 类的基类，但不是虚基类 参考答案 A

二进制文件与文本文件不同的是()。

A 二进制文件中每字节数据都没有用ASCII码表示 B 二进制文件包含了ASCII码控制符 C 二进制文件一般以字符‘\0’结束 D

二进制文件用字符endl表示行的结束 参考答案 A

`cin`是I/O 流库预定义的()。

A 类 B 对象 C 包含文件 D 常量 参考答案 B

当要建立一个文件，并从其中读数据和向其中写数据时，应该建立下面哪种流类的对象（ ）。

A ifstream B ofstream C fstream D ios 参考答案 C

当要用I/O流操作文件时，需要包含（ ）文件。

A iostream B fstream C iostream D cmath 参考答案 B

下面关于C++流的叙述中，正确的是（ ）。

A cin是一个输入流对象 B 可以用ifstream定义一个输出流对象 C 执行语句序列char *y="Happy new year"; cout<<y;时，输出为：

Happy D 执行语句序列char x[80];cin.getline(x,80);时，若键入Happy new year则x中的字符串是"Happy" 参考答案 A

下列输出字符'A'的方法中，错误的是（ ）。 A cout<<put('A') ; B cout<<' A' ; C cout. put('A') ; D char A=' A' ; cout<<A;

参考答案 A

C++流中的重载运算符<<,它是一个（ ）

A 用于输出操作的成员函数。 B 用于输入操作的成员函数。 C 用于输出操作的非成员函数。 D 用于输入操作的非成员函数。 参考答案 B

磁盘文件操作时，打开磁盘文件的访问方式常量中，（ ）是以追加方式打开文件的。

A in B out C app D ate 参考答案 C

下列关于函数read的正确描述是（ ）。

A 该函数只能用于文本文件的操作中。 B 该函数只能用来从键盘输入中获取字符串。 C 该函数所获取的字符多少是不受限制的。 D 该函数只能按规定读取所指定的字符数。 参考答案 D

当使用ifstream流定义一个流对象并打开一个磁盘文件时，文件的隐含打开方式为（ ）。

A ios::in B ios::out C ios::trunk D ios::binary 参考答案 A

在进行完任何C++流的操作后，都可以用C++流的有关成员函数检测流的

状态；其中只能用于检测输入流状态的操作函数 名称是 () A fail
B eof C bad D good 参考答案 A

在C++中，所有关于流的类都是由 ios 类派生出来的。

[参考答案] 正确

cin 和cout 是C++中的输入输出库函数。 [参考答案] 错误

因为cin 和cout 是 C++的默认输入/输出流，因此若要使用cin 和cout 不必包含什么头文件。 [参考答案] 错误

ios 是虚基类，不能用于定义对象。 [参考答案] 正确

在 C++ 中，符号 “<<” 和 “>>” 被重载成为输入和输出操作符。

[参考答案] 错误

C++语言中，根据数据在文件中的编码形式不同，把文件分为文本文件和__1__两种 。 参考答案1：二进制文件 或 二进制

若在I/O流的输出中使用控制符setfill()设置填充字符，应包含的头文件是__1__。 参考答案1： iomanip.h 或 iomanip

C++在类ios中定义了输入输出格式控制符，它是一个__1__。该类型中的每一个量对应两个字节数据的一位，每一个位代表一 种控制，如要取多种控制时可用__2__运算符来合成。

参考答案1： 枚举类型2： 或 |

采用输入输出格式控制符，其中有参数的，必须要求包含__1__头文件。 参考答案1： iomanip.h 或 iomanip

C++根据文件内容的数据格式可分为两类：文本文件和二进制文件，前者存取的最小信息单位为__1__，后者为__2__。

参考答案1： 字符2： 字节

cin通过重载__1__执行输入，而cout，cerr和clog通过重载__2__执行输出。 参考答案1： >>2： <<

从流中取得数据称为__1__，用符号__2__表示

参考答案1：提取操作 或 提取2：>>

向流中添加数据称为__1__，用符号__2__表示。

参考答案1：插入操作 或 插入2：<<

类__1__是所有基本流类的基类，它有一个保护访问限制的指针指向类__2__，其作用是管理一个流的缓冲区。

参考答案1：ios2：streambuf

下面的程序把一个整数文件中数据乘以10后写到另一文件中。在空白处填入合适的内容：

```
#include <iostream.h>
#include <fstream.h>
void main()
{
    char f1[20],f2[20];
    cout<<"输入源文件名:";          cin.getline(f1,20);
    ifstream input(__1__
);
    if(!input){          cerr<<"源文件不存在"<<endl;    return;
}

    cout<<"输入目标文件名:";          cin.getline(f2,20);
    ofstream output(f2,ios::noreplace);
    if(!output)    {          cerr<<"目标文件已存在"<<endl;
return;          }

    int number;
    while(__2__
) output<<number*10<<' \t' ;
```

```
input.close();    output.close();    }
```

参考答案1: f1, ios::nocreate 2: input>>number

下列程序将结构体变量tt中的内容写入D盘上的date.txt文件。在空白处填入合适的内容:

```
#include <fstream.h>

struct date
{
    int year, month, day;    };

void main()
{
    date tt={2002, 2, 12};

    ofstream(__1__

);

    outdate.open("d:\\date.txt", ios::binary);

    if (!outdate)
        { cerr << "\n 文件不能打开" << endl ;        return ; }

    outdate.write(__2__);    }
```

参考答案1: outdate 2: (char*) &tt, sizeof(tt)

以下程序是将文本文件data.txt中的内容读出并显示在屏幕上。在空白处填入正确的内容:

```
#include<fstream.h>

void main()
{
    char buf[80];

    ifstream me("e:\\exercise\\data.txt");

    while(__1__

)
```

```
{
    me.getline(buf, 80);
    cout<<__2__<<endl;    }    }
```

参考答案1: !me.eof() 2: buf

对二进制文件的I/O操作使用read和__1__成员函数来完成。

参考答案1: write

C++对文件读写实际上受到文件定位指针的控制，输入流的指针也称为__1__，每一次提取从该指针所指位置开始。输出流的 指针也称为__2__，每一次插入也从该指针所指位置开始。每次操作后自动将指针向文件尾移动。如果能任意向前向后移动该 指针，则可实现随机读写。参考答案1: 读指针 2: 写指针

C++把每一个文件都看成一个有序的字节流，并以__1__结束。

参考答案1: 文件结束符

在C++中，打开一个文件就是将一个文件与一个__1__建立关联；关闭一个文件就是取消这种关联。参考答案1: 流

已有定义

```
int a=20;
```

执行语句cout<<hex<<a<<endl;的输出结果是

__1__。

参考答案1: 14

语句 cout.put(65+44); 的输出结果是__1__。参考答案1: m

重载的流操作符经常定义为类的__1__函数。参考答案1: 友元

表达式 cout<<' \n' 还可以表示为

__1__。

参考答案1: cout<<endl

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    char s[25]="Programming language";
    ofstream f1("DATA.TXT");
    f1<<"C++ Programming";
    f1.close();
    ifstream f2("DATA.TXT");
    if(f2.good())    f2>>s;
    f2.close();
    cout<<s;
    return 0;
}
```

执行上面的程序将输出

__1__。

参考答案1: C++

在C++中输入输出是通过__1__实现的。 **参考答案1:** 流

C++中有输入文件流、输出文件流和__1__3种。

参考答案1: 输入输出文件流

在C++中，打开一个文件就是将一个文件与一个__1__建立关联；关闭一个文件就是取消这种关联。 **参考答案1:** 流

C++的文本文件格式，存取的最小信息单位为__1__。

参考答案1: 字节

要使用如setw, fixed等操作符时, 需要包含文件__1__。

参考答案1: iomanip; iomanip.h

cout的默认输出对象是__1__。参考答案1: 屏幕

若要想继续标准输入/输出则头文件必须包含__1__文件。

参考答案1: iostream.h; iostream

在C++流类库中, 根基类为__1__。参考答案1: ios类

流插入运算符是__1__。参考答案1: <<

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

```
void main()
```

```
{    int n=0;
```

```
    char ch;
```

```
    cout<<"input:";
```

```
    while((ch=cin.get())!=EOF)
```

```
        n=n+1;
```

```
    cout<<n<<endl;
```

```
}
```

当从键盘上输入:

This is a book.<Enter>

<Ctrl+Z>

时, 输出结果是 () 参考答案 16 3.

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

```
#include <iomanip.h>
```



```

void main(void)
{
    int i;
    for (i = 0; i < 3; i++)
    {
        cout.width(5);
        cout << i;
    }
    cout<<endl;
    cout.setf(ios::left);
    for (i = 0; i < 3; i++)
    {
        cout.width(5);
        cout << i;
    }
}

```

程序的运行结果是 **参考答案** 0 1 2 0 1 2

```

#include <iostream.h>
#include <iomanip.h>
void main() {
    cout.setf(ios::fixed);
    cout.precision(3);
    cout.fill('*');
}

```

```
cout<<setw(8)<<12.345;

cout<<setw(10)<<34.567<<endl ;

}
```

程序的运行结果是： **参考答案** **12.345***34.567

下列叙述中正确的是

A 栈是“先进先出”的线性表 B 队列是“先进后出”的线性表 C 循环队列是非线性结构 D 有序线性表既可以采用顺序存储结构，也可以采用链式存储结构 **参考答案** A

已知函数FA调用FB，若要把这两个函数定义在同一个文件中，则 A FA必须定义在FB之前 B FB必须定义在FA之前 C 若FA定义在FB之后，则FA的原型必须出现在FB的定义之前 D 若FB定义在FA之后，则FB的原型必须出现在FA的定义之前 **参考答案** D

数据库管理系统是

A 操作系统的一部分 B 在操作系统支持下的系统软件 C 一种编译系统 D 一种操作系统 **参考答案** B

在E-R图中，用来表示实体联系的图形是

A 椭圆形 B 矩形 C 菱形 D 三角形 **参考答案** C

下列选项中不属于结构化程序设计原则的是

A 可封装 B 自顶向下 C 模块化 D 逐步求精 **参考答案** A

软件设计中划分模块的一个准则是

A 低内聚低耦合 B 高内聚低耦合 C 低内聚高耦合 D 高内聚高耦合 **参考答案** A

软件详细设计产生的图如下：

begin

X=0

endl

该图是：

A N-S图 B PAD图 C 程序流程图 D E-R图 参考答案 C

算法的空间复杂度是指

A 算法在执行过程中所需要的计算机存储空间 B 算法所处理的数据量
C 算法程序中的语句或指令条数 D 算法在执行过程中所需要的临时工作单元数 参考答案 A

有如下程序：

```
#include<iostream>

using namespace std;

class ONE {
    int c;
public:
    ONE(): c(0) { cout<<1; }
    ONE(int n): c(n) { cout<<2; }
};

class TWO {
    ONE one1;
    ONE one2;
```

```
public:
TWO(int m) : one2(m) { cout<<3; }
};

int main() {
TWO t(4);
return 0;
}
```

运行时的输出结果是

A 3 B 23 C 123 D 213 参考答案 D

有如下程序段：

```
int i=4; int j=1;

int main() {
int i=8, j=i;
cout<<i<<j<<endl;
}
```

运行时的输出结果是

A 44 B 41 C 88 D 81 参考答案 B

有如下程序段：

```
int i=5;
while (int i=0) {cout<<'*' ; i--; }
```

运行时输出 “*” 的个数是

A 0 B 1 C 5 D 无穷 参考答案 B

某二叉树有5个度为2的结点以及3个度为1的结点，则该二叉树中共有

__1__个结点。 **参考答案1**: 14

人员基本信息一般包括：身份证号，姓名，性别，年龄等。其中可以作为主关键字的是__1__。 **参考答案1**: 身份证号

软件开发过程主要分为需求分析、设计、编码与测试四个阶段，其中__1__阶段产生“软件需求规格说明书”。

参考答案1: 需求分析

在数据库技术中，实体集之间的联系可以是一对一或一对多或多对多的，那么“学生”和“可选课程”的联系为__1__。

参考答案1: 多对多

程序流程图中的菱形框表示的是__1__。 **参考答案1**: 逻辑条件

在E-R 图中，图形包括矩形框、菱形框、椭圆框。其中表示实体联系的是__1__。 **参考答案1**: 菱形

软件测试可分为白盒测试和黑盒测试。基本路径测试属于__1__测试。

参考答案1: 白盒

数据库系统的核心是__1__。 **参考答案1**: 数据库管理系统

有一个学生选课的关系，其中学生的关系模式为：学生（学号，姓名，班级，年龄），课程的关系模式为：课程（课号，课程 名，学时），其中两个关系模式的键分别是学号和课号，则关系模式选课可定义为：选课（学号，__1__，成绩）。 **参考答案1**: 1

软件是__1__、数据和文档的集合。 **参考答案1**: 11

设某循环队列的容量为50，如果头指针front=45（指向队头元素的前一位置），尾指针rear=10（指向队尾元素），则该循环队 列中共有__1__个元素。 **参考答案1**: 1

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```

int row(9);
int i, j, n;
n=row/2+1;
for(i=1;i<=n;i++)
{
    for(j=1;j<=n-i;j++)
        cout<<' ';
    for(j=1;j<=2*i-1;j++)
        cout<<' *' ;
    cout<<endl;
}
for(i=1;i<=n-1;i++)
{
    for(j=1;j<=i;j++)
        cout<<' ';
    for(j=1;j<=row-2*i;j++)
        cout<<' *' ;
    cout<<endl;
}
return 0;
}

```

参考答案:

*

*

下列关于布尔型数据与数值型数据的转换说法中，错误的是()。

A 布尔量true当数值使用时相当于1； B 布尔量false当数值量使用时相当于0； C 数值量-1当布尔量使用时相当于false； D 数值量0当布尔量使用时相当于false； 参考答案 C

以下程序段的结果是()。

```
int m=0, n=0;
if ( ++m || ++n )
cout << m << ", " << n << endl;
```

A 0,0 B 1,1 C 1,0 D 0,1 参考答案 C

以下程序段的结果是()。

```
int m=0, n=0;
if ( m++ && n++ )
    m++;
cout << m << ", " << n << endl;
```

A 0,0 B 1,1 C 1,0 D 2,1 参考答案 C

以下程序段的结果是()。

```
int m=0, n=0;
```

```
if ( ++m && ++n )
```

```
cout << m << ", " << n << endl;
```

A 0,0 B 1,1 C 1,0 D 0,1 参考答案 B

有循环结构程序段:

```
for(i=1;i<10;i++)
```

```
cout<<i<<endl;
```

```
cout<<i;
```

下列说法错误的是【】。

A 循环体语句被执行了9次; B 循环体只包含一个语句; C 第2个cout语句输出i值为9; D 1, 2, 3, 4, 5, 6, 7, 8, 9, 10都被输出。 参考答案 C

已知int i, x, y; 在下列选项中错误的是()。 A if (x && y) i++; B if (x = y) i--; C if (x y) i--; D if (x + y) i++; 参考答案 C

假设i=2, 执行下列语句后i的值为()。

```
switch(i)
```

```
{
```

```
    case 1: i++;
```

```
    case 2: i--;
```

```
    case 3: ++i; break;
```

```
    case 4: --i;
```

```
    default: i++;
```

```
}
```

A 1 B 2 C 3 D 4 参考答案 B

以下为死循环的程序段是（ ）。

A for (int x=0; x<3; x--) x+=2; B int k = 0;do
{ ++k; } while (k>=0);C int a = 5; while (a) { a--; }
D int i = 3; for (; i; i--);**参考答案** B

若有如下的函数定义：

```
int sum( int a, float b)
{ return a+(int)b; }
```

则下列函数原型不正确的是（ ）。

A int sum(int, float); B int sum(int x, float y); C int
sum(int b, float a); D int sum(int a=10, float b); **参考答
案** D

有以下程序段：

```
#include <iostream.h>
#define F(X,Y) (X)*(Y)
void main()
{
    int a=3, b=4;
    cout<< F(a++, b++)<<endl;
}
```

程序执行后，输出的结果是__1__。

参考答案1： 12

有以下程序段：

```
#include <iostream.h>
#define EVEN(x) x%2==0?1:0
```

```
void main( )  
{  
    if(EVEN(9+1)) cout<<0;  
    else cout<<1;  
}
```

程序执行后，输出的结果是__1__。

参考答案1： 1

有以下程序段：

```
#include <iostream.h>  
#define P 3  
int F(int x)  
{    return(P*x*x); }  
void main()  
{    cout<<F(3+5); }
```

程序执行后，输出的结果是__1__。

参考答案1： 192

有以下程序：

```
#include <iostream.h>  
#define ADD(x) (x)+(x)  
void main()  
{  
    int a=4, b=6, c=7, d=ADD(a+b)*c;  
    cout<<d;
```

```
}
```

程序执行后输出的结果是__1__。

参考答案1: 80

在定义一个函数时，若只允许函数体访问一个形参的值，不允许修改它的值，则应把该形参说明为常量，即在形参说明的前面 加上__1__关键字进行修饰。**参考答案1:** const

在进行函数调用时，将把实际参数的值传递给值参数，把实际参数的__1__传递给引用参数。**参考答案1:** 地址

当调用一个函数时，整个调用过程分为三步进行，第一步是参数传递，第二步是函数体执行，第三步是__1__，即返回到函数调用 表达式的位置。**参考答案1:** 返回

阅读以下程序：

```
void func1(int i);
void func2(int i);
char st[]="hello,friend!";
void func1(int i)
{
    cout<<st[i];
    if(i<3) { i+=2; func2(i); }
}
void func2(int i)
{
    cout<<st[i];
    if(i<3) { i+=2; func1(i); }
}
```

```

void main()
{
    int i=0;
    func1(i);
    cout<<endl;
}

```

写出执行后的输出结果。

参考答案 hlo

有以下程序：

```

int  abc(int u,int v);
void main ()
{
    int a=24, b=16, c;
    c=abc(a, b);
    cout<<c;
}

int abc(int u, int v)
{
    int  w;
    while ( v )
    {   w=u%v;   u=v;   v=w ;   }
    return u;
}

```

写出输出结果。

参考答案 8

有以下程序：

```
#include<iostream>
using namespace std;
func( int a, int b)
{
    int c;
    c=a+b;
    return c;
}
void main()
{
    int x=6, y=7, z=8, r;
    r=func( (x--, y--, x+y), z--);
    cout<<r<<endl;
}
```

写出输出结果。

参考答案 19

以下能正确定义一维数组的选项是（ ）。

A int a[5]={0, 1, 2, 3, 4, 5}; B char a[]={0, 1, 2, 3, 4, 5}; C char a={' A', ' B', ' C' }; D int a[5]=" 0123" ; 参考答案 B

以下叙述中错误的是（ ）。

A 对于double类型数组，不可以直接用数组名对数组进行整体输入或输出 B 数组名代表的是数组所占存储区的首地址，其值不可改变 C 在程序执行中，当数组元素的下标超出所定义的下标范围时，系统将给出“下标越界的出错信息” D 可以通过赋初值的方式确定数组元素的个数 参考答案 C

有以下程序：

```
void main()
{
    int a[]={2,4,6,8,10}, y=0, x, *p;
    p=&a[1];
    for(x=1; x<3; x++)
        y+=p[x];
    cout<<y<<endl;
}
```

程序运行后的输出结果是（ ）。

A 10 B 11 C 14 D 15 参考答案 C

有以下程序：

```
void sort(int a[], int n)
{
    int i, j, t;
    for(i=0; i<n-1; i++)
        for(j=i+1; j<n; j++)
            if(a[i]<a[j])
                { t=a[i]; a[i]=a[j]; a[j]=t; }
```

```

}

void main()
{
    int aa[10]={1,2,3,4,5,6,7,8,9,10}, i;
    sort(aa+2, 5);
    for(i=0; i<10; i++)
        cout<<aa[i]<<" ";
}

```

程序运行后的输出结果是 ()。

A 1 2 3 4 5 6 7 8 9 10 B 1 2 7 6 3 4 5 8 9 10 C 1 2 7 6 5 4 3
8 9 10 D 1 2 9 8 7 6 5 4 3 10 参考答案 C

有以下程序：

```

#define N 20

void fun(int a[], int n, int m)
{
    for(int i=m; i>=n; i--)
        a[i+1]=a[i];
}

void main()
{
    int a[N]={1,2,3,4,5,6,7,8,9,10}, i;
    fun(a, 2, 9);
    for(i=0; i<5; i++)

```

```
        cout<<a[i];  
    }
```

程序运行后的输出结果是（ ）。

A 10234 B 12344 C 12334 D 12234 参考答案 C

以下语句中存在语法错误的是（ ）。

A char ss[6][20]; ss[1]="right?"; B char ss[][20]={"right?";
C char *ss[6]; ss[1]="right?"; D char *ss[]={ "right?"}; 参考答
案 A

有定义语句:

```
char s[10];
```

若要从终端给s输入5个字符，错误的输入语句是（ ）。

A cin.getline(s,6); B cin>>s+1; C cin.gets(s,6); D cin>>s[1];
参考答案 D

设有定义:

```
char s[81];
```

```
int i=0;
```

以下不能将一行（不超过80个字符）带有空格的字符串正确读入的语句或语句组是（ ）。

A cin.getline(s,80); B do { s[i]=cin.get(); } while (s[i++]!=
' #');s[++i]=' \0' ; C cin>>s; D while ((s[i++]=cin.get())!=
' #');s[i]=' \0' ; 参考答案 C

设有以下程序段，若先后输入： Country ✓ Side ✓

则其运行结果为（ ）。

```
void main()
```

```
{
```



```

char c1[60], c2[30];
int i=0, j=0;
cin>>c1>>c2;
while(c1[i]!=' \0' ) i++;
while(c2[j]!=' \0' ) c1[i++]=c2[j++];
c1[i]=' \0' ;
cout<<c1<<endl;
}

```

ACountrySide BCountry C Side D Country Side 参考答案 A

有以下程序：

```

void main()
{
    char s[]="abcde";
    s+=2;
    cout<<s[0]<<endl;
}

```

程序运行后的输出结果是（ ）。

A 输出字符a的ASCII B 输出字符c的ASCII码 C 输出字符c D 程序出错
 参考答案 D

以下程序段中，不能正确赋字符串（编译时系统会提示错误）的是
 （ ）。

A char s[10]="abcdefg"; B char t[10]="abcdefg", *s=t; C char s[10]; s="abcdefg"; D char s[10]; strcpy(s, "abcdefg"); 参考答案 C

以下能正确定义字符串的语句是（ ）。

A char str[]={ ' \064' }; B char str="\x43"; C char str='
' ; D char str[]="\0"; 参考答案 D

设有定义：

```
char p[]={ ' 1' , ' 2' , ' 3' }, *q=p;
```

以下不能计算出一个char型数据所占字节数的表达式是（ ）。

A sizeof(p) B sizeof(char) C sizeof(*q) D sizeof(p[0]) 参考答
案 A

有以下程序段：

```
char a[10]="abcd";  
cout<<strlen(a)<<"", "<<sizeof(a)<<endl;
```

则执行后的输出结果是（ ）。

A 7, 4 B 4, 10 C 8, 8 D 10, 10 参考答案 B

若有定义语句：

```
char s[10]="1234567\0\0";
```

则strlen(s)的值是（ ）。

A 7 B 8 C 9 D 10 参考答案 A

定义数组长度时，其“元素个数”允许的方式是（ ）。

A 整型常量 B 整型表达式 C 整型常量或整型表达式 D 任何类型的表
达式 参考答案 A

下面描述中不正确的是（ ）。

A 字符型数组中可以存放字符串 B 可以对字符型数组进行整体输入、
输出 C 可以对整型数组进行整体输入、输出 D 不能在赋值语句中通过
赋值运算符“=”对字符型数组进行整体赋值 参考答案 C

在定义int x[3][2]以后，对x元素的引用正确的是（ ）。

A x[0, 0] B x[3][1] C x[2][2] D x[0][0] 参考答案 D

已知int型二维数组a[3][4]，则与数组元素的指针访问方式*(*
(a+1)+2)等价的下标访问方式表示为__1__。

参考答案1: a[1][2]

有以下定义：

```
char s1[20]="Good!";
```

```
char s2[15]="AB";
```

则strlen(strcpy(s1,s2))的值为：__1__。

参考答案1: 2

定义了int型二维数组a[6][7]后，数组元素a[3][4]前的数组元素个数为__1__ 参考答案1: 25

数组元素a[I]是该数组中的第__1__个元素(个数从1开始计数)。 参考答案1: I+1 或 1+I

在定义时对数组的每一个元素赋值叫数组的__1__。

参考答案1: 初始化

有以下程序段：

```
char b[]="Hello, you";
```

```
b[2]=0;
```

```
cout<<b<<endl;
```

则执行后，其输出结果是__1__。

参考答案1: He

假定int类型变量占用4个字节，其有定义：int x[10]={0,2,4};，则数组x在内存中所占字节数是__1__。

参考答案1: 40

若有以下说明:

```
int a[12]={1,2,3,4,5,6,7,8,9,10,11,12};
```

```
char c=' a' , g=' e' ;
```

则表达式a[g-c] 的值是__1__。

参考答案1: 5

若有以下定义语句:

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

则表达式 “x[1][1]*x[2][2]” 的值是__1__。

参考答案1: 18

有定义int n[5][6], 则元素n[1][1]后(不包括n[1][1])第10个元素是__1__。 参考答案1: n[2][5]

若有以下定义:

```
char s[ ]= “\\141\\141abc\\t “;
```

则函数strlen(s)的值为__1__。

参考答案1: 9

以下程序从键盘输入N个整数, 用选择法排序法按从小到大的次序输出。

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

```
#define N 10
```

```
void main( )
```

```
{
```

```
    int i, j, min, temp, a[N];
```

```
    for ( i=0; i<N; i++)          //输入N个数
```

```

        cin>>__1__ ;
for ( i=0; __2__ ; i++)
{
    min=i;
    for ( j=i; j<N; j++)
        if(a[min]>a[j]) __3__ ;
    temp=a[i]; a[i]=a[min]; a[min]=temp;
}

for (i=0;i<N;i++)          //输出排序后的数
    cout<<a[i]<<"\t";
cout<<endl;
}

```

参考答案1: a[i] 2: i<N-1 或 i<=N-2 或 N-1>i 或 N-2>=I
 3: min=j 或 min=j;

下面程序将输出两个字符串中对应字符相等的字符。请填空。

```

#include<iostream.h>

void main()
{
    char   x[ ]= "language";
    char   y[ ]= "llngga";
    int    i=0;
    while (x[i] || y[i])
    {

```

```

        if ( __1__ )    cout<<x[i];
        __2__ ;
    }
    cout<<endl;
}

```

参考答案1: x[i]==y[i] 或 y[i]==x[i]

2: i++ 或 ++i 或 i+=1 或 i=i+1

以下程序将字符数组a中下标值为偶数的元素从小到大排列，其他元素不变。请填空。

```

#include<iostream.h>
#include <string.h>
void main()
{
    char  a[]="clanguage", t;
    int  i, j, k;
    cout<<a<<endl;
    k=strlen(a);
    for ( i=0; i<=k-2; i+=2)
        for (j=i+2; j<=k; __1__ )
            if ( __2__ )
                {  t=a[i]; a[i]=a[j]; a[j]=t;  }
    cout<<a<<endl;
}

```

参考答案1: j=j+2 或 j+=2 或 j++, j++

2: a[i]>a[j] 或 a[j]<a[i]

从键盘输入由5个字符组成的单词，判断此单词是不是hello，并显示结果。

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

```
void main()
```

```
{
```

```
    char str[ ]={"hello"}, str1[5];
```

```
    __1__ ;
```

```
    for(int i=0; i<5; i++)          //输入由5个字符
```

```
        __2__ ;
```

```
    flag=0;
```

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

```
        if ( __3__ )
```

```
        { flag=1; break; }
```

```
    if(flag)    cout<<"输入的不是hello";
```

```
    else    cout<<"输入的是hello";
```

```
    cout<<endl;
```

```
}
```

参考答案1: int flag; 或 int flag

2: cin>>str1[i]; 或 cin>>str1[i]

3: str[i]!=str1[i] 或 str1[i]!=str[i]

以下程序把一个整数转换成二进制数，所得二进制数的每一位放在一维数组中，输出此二进制数。完成此程序，注意：二进制 数的最低位在

数组的第0个元素中。

```
#include <iostream.h>

void main()
{
    int  b[16], x, k=-1, r, i;
    cout<<"输入一个十进制数到x:";  cin>>x;
    do
    {
        r=x%__1__;
        b[++k]=r;
        x/=__2__;
    } while(x>=1) ;
    for ( i=k; __3__ ; i--)          //输出二进制数
        cout<<b[i];
    cout<<endl;
}
```

参考答案1: 2 或 2; 2: 2 或 2;

3: i>=0 或 0<=i

以下程序的功能是：从键盘上输入若干个学生的成绩，计算出平均成绩，并输出低于平均分的学生成绩，用输入负数结束输入。请填空。

```
#include<iostream.h>

void main()
{
```



```

float  x[100], sum=0, ave, a;
int    n=0, i;
cout<<"Enter mark: \n";  cin>>a;
while (a>=0 && n<100)
{
    sum+=__1__; x[n]= a;
    n++; cin>>a;
}
ave=__2__;
cout<<"Output: \n";
cout<<ave<<endl;
for  (i=0; i<n; i++ )
    if ( __3__ )  cout<<x[i]<<" ";           //输出低于平均分的
    学生成绩
}

```

参考答案1: a 或 a; 2: sum/n 或 sum/n;

3: x[i]<ave 或 ave>x[i]

以下程序分别在a数组和b数组中放入an+1和bn+1个由小到大的有序数，程序把两个数组中的数按由小到大的顺序归并到c数组 中。

```

#include<iostream.h>

void main()
{
    int a[10]={1,2,5,8,9,10}, an=5;
    int b[10]={1,3,4,8,12,18}, bn=5;

```

```

int i, j, k, c[20];
i=j=k=0;
while(i<=an || j<=bn)
    if(a[i]<b[j]) { c[k]=__1__; k++; __2__; }
    else { c[k]=__3__; k++; j++; }
for(i=0; i<k; i++)
    cout<<c[i]<<"\t";
cout<<endl;
}

```

参考答案1: a[i] 或 a[i];

2: i++ 或 i=i+1 或 i+=1 或 ++i

3: b[j]; 或 b[j]

请写出下面程序运行的结果。

```

#include<iostream.h>
void main()
{
    int a[3][3]={1,2,3,4,5,6,7,8,9}, i, s1=0, s2=1;
    for(i=0; i<=2; i++)
    {
        s1=s1+ a[i][i];
        s2=s2*a[i][i];
    }
    cout<<s1<<s2<<endl;
}

```

```
}
```

参考答案 1545

阅读以下程序：

```
void reverse(int a[ ], int n)
{
    int i, t;
    for(i=0; i<n/2; i++)
        { t=a[i]; a[i]=a[n-1-i]; a[n-1-i]=t; }
}

main()
{
    int b[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int i, s=0;
    reverse(b, 8);
    for ( i=6; i<10; i++ ) s+=b[i];
    cout<<s<<endl;
}
```

写出执行后的输出结果。

参考答案 22

有如下程序：

```
#include<iostream>
using namespace std;
int fun( int h )
```

```

{
    static int a[3]={1, 2, 3};
    int k;
    for ( k=0; k<3; k++ )    a[k]+=a[k]-h;
    for ( k=0; k<3; k++ )    cout<<a[k];
    return ( a[h] );
}
void main()
{
    int t=1;
    fun(fun(t));
}

```

写该程序的输出结果。

参考答案 135-137

有以下程序：

```

void sum(int *a)
{ a[0]=a[1]; }
void main()
{
    int aa[10]={1,2,3,4,5,6,7,8,9,10}, i;
    for(i=2; i>=0; i--)
        sum(&aa[i]);
    cout<<aa[0]<<endl;
}

```

```
}
```

执行后的输出结果是()。

A 4 B 3 C 2 D 1 参考答案 A

主调函数中的两个变量a和b，要求调用函数交换a、b的值，返回交换结果，则以下正确的函数是()。

A funa(int *x, int *y) {int *p;*p=*x; *x=*y; *y=*p;

} B funa(int x, int y) {int t;t=x; x=y; y=t;}

C funa(int *x, int *y) {*x=*y; *y=*x;} D funa(int *x, int *y)

{*x=*x+*y; *y=*x-*y; *x=*x-*y;} 参考答案 D

有以下程序段：

```
void main()
```

```
{
```

```
    char s[]="159", *p;
```

```
    p=s;
```

```
    cout<<*p++<<" ";
```

```
    cout<<*p++<<endl;
```

```
}
```

程序运行后输出结果是()。

A 1 5 B 1 6 C 1 2 D 5 9 参考答案 A

有以下程序：

```
void main()
```

```
{
```

```
    int a=7, b=8, *p, *q, *r;
```

```
    p=&a;
```

```

    q=&b;

    r=p;

    p=q;

    q=r;

    cout<<*p<<*q<<a<<b<<endl;

}

```

程序执行后的输出结果是()。

A 8787 B 7878 C 8778 D 7887 参考答案 C

有以下程序：

```

void main()
{
    int a=7,b=8,*p,*q,*r;

    p=&a;

    q=&b;

    r=p;

    p=q;

    q=r;

    cout<<*p<<*q<<a<<b<<endl;

}

```

程序执行后的输出结果是()。

A 8787 B 7878 C 8778 D 7887 参考答案 C

有以下程序：

```

void main()

```

```

{
    int a[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}, *p=a+5, *q=NULL;
    *q=*(p+5);
    cout<<*p<<*q<<endl;
}

```

程序运行后的输出结果是()。

A 运行后报错 B 66 C 611 D 510 参考答案 A

设有定义：

```
int n=0, *p=&n, **q=&p;
```

则以下选项中正确的赋值语句是()。

A p=1; B *q=2; C q=p; D *p=5; 参考答案 D

有以下函数：

```

fun(char *a, char *b)
{
    while ((*a!=''\0') && (*b!=''\0') && (*a==*b))
    {
        a++;
        b++;
    }
    return (*a-*b);
}

```

该函数的功能是()。

A 计算a和b所指字符串的长度之差 B 将b所指字符串连接到a所指字符

串中 C 将b所指字符串连接到a所指字符串后面 D 比较a和b所指字符串的大小 参考答案 D

设有定义：

```
int n1=0, n2, *p=&n2, *q=&n1;
```

以下赋值语句中与n2=n1语句等价的是()。

A *p=*q; B p=q; C *p=&n1; D p=*q; 参考答案 A

有以下程序：

```
void main()
{
    int a[10]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, *p=&a[3], *q=p+2;
    cout<<*p*q<<endl;
}
```

程序运行后的输出结果是()。

A 16 B 10 C 8 D 6 参考答案 B

有以下程序：

```
void main()
{
    int a[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 0}, *p;
    for (p=a; p<a+10; p++)
        cout<<*p<<" "; }
```

程序运行后的输出结果是()。

A 1 2 3 4 5 6 7 8 9 0 B 2 3 4 5 6 7 8 9 10 1 C 0 1 2 3 4 5 6 7 8 9 D 1 1 1 1 1 1 1 1 1 1 参考答案 A

有以下程序：


```

void swap1(int c0[], int c1[])
{
    int t;
    t=c0[0]; c0[0]=c1[0]; c1[0]=t;
}

void swap2(int *c0, int *c1)
{
    int t;
    t=*c0; *c0=*c1; *c1=t;
}

void main()
{
    int a[2]={3, 5}, b[2]={3, 5};
    swap1(a, a+1);
    swap2(&b[0], &b[1]);
    cout<<a[0]<<a[1]<<b[0]<<b[1]<<endl;
}

```

程序运行后的输出结果是（ ）。

A 3553 B 5335 C 3535 D 5353 **参考答案 D**

有以下程序：

```

void f(int *x, int *y)
{
    int t;

```

```

        t=*x; *x=*y; *y=t;
    }
void main()
{
    int a[8]={1,2,3,4,5,6,7,8}, i, *p, *q;
    p=a;
    q=&a[7];
    while(p<q)
    { f(p, q); p++; q--; }
    for(i=0; i<8; i++)
        cout<<a[i];
}

```

程序运行后的输出结果是（ ）。

A 82345671 B 56781234 C 12345678 D 87654321 **参考答案 D**

设有定义：

```
char *c;
```

以下选项中能够使字符型指针c正确指向一个字符串的是（ ）。

A char str[]="string"; c=str; B cin>>c; C cin.getline(c,10);
D *c="string"; **参考答案 A**

设有以下定义和语句：

```
char str[20]="Program", *p;
```

```
p=str;
```

则以下叙述中正确的是（ ）。

A *p与str[0]中的值相等 B str与p类型完全相同 C str数组长度和p所指向的字符串长度相等 D 数组str中存放的内容和指针变量p中存放的内容相同 参考答案 A

有以下程序：

```
void main()
{
    char s[10]={"aeiou"},*ps;
    ps=s;
    cout<<char(*ps+4)<<endl;
}
```

程序运行后的输出结果是（ ）。

A a B e C u D 元素s[4]的地址 参考答案 B

已知int a[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9}，则不能表示数组元素a[2][1]的地址是（ ）。

A &a[2][1] B *(a[2]+1) C a[2]+1 D *(a+2)+1 参考答案 B

已知函数f的原型为：

```
void f (int &a, char *b);
```

变量s、t的定义是：

```
int s;
```

```
char t[ ]="ABCD";
```

把s和t分别作为第一参数和第二参数来调用函数f，正确的调用语句是（ ）。

A f(&s, &t); B f(&s, t); C f(s, t); D f(s, &t); 参考答案 C

有以下声明：

```
int fun(int x), (*p)(int);
```

则使指针变量p指向fun函数入口的语句为:

```
p=__1__;
```

参考答案1: fun 或 fun;

在C++中, 一个指针类型的变量占用内存的__1__个字节的存储空间。 参考答案1: 4

假设有以下定义:

```
int a, x=5, *p=&x;
```

则执行a = *p++;语句后, a的值为__1__。

参考答案1: 5

下面程序的功能是将字符串s的所有字符传送到字符串t中, 要求每传递三个字符后再存放一个空格, 例如字符串s为"abcdefg", 则字符串t为"abc def g", 请选择填空。

```
#include <iostream>
```

```
using namespace std;
```

```
void main( )
```

```
{
```

```
    int j, k=0;
```

```
    char s[60]="abcdefg", t[100], *p;
```

```
    p=s;
```

```
    while(*p)
```

```
    {
```

```
        for (j=1; j<=3 && *p; j++, k++, __1__) t[k]=*p;
```

```
        if ( __2__ ) { t[k]=32; k++; } //32为空格的
```

ASCII码值

```
    }  
    t[k]=0;  
    cout<<t<<endl;  
}
```

参考答案1: p++ 或 p=p+1 或 ++p 或 p+=1

2: j==4 或 4==j

有下面的程序段:

```
int a, b, k=4, m=6, *p=&k, *q=&m;  
a=p==&m;  
b=(-*p)/(*q)+7;  
cout<<a<<b<<endl;
```

则输出的结果是__1__。

参考答案1: 07

有下面的程序段

```
char *s="abcde";  
s+=2;  
cout<<s<<endl;
```

则运行结果是__1__。

参考答案1: cde

若有以下定义和语句:

```
int s[4][5]={1, 2, 3, 4, 5, 6, 7, 8, 9}, (*ps)[5];  
ps=s;
```

```
cout<<*(*ps+6);
```

则执行后的输出结果是__1__。

参考答案1: 7

若指针p已正确定义，要使p指向10个连续的char型存储单元，使用语句：

```
p=new __1__;
```

参考答案1: char[10] 或 char[10];

执行下面程序段：

```
char *s="\ta\018b\0c";
```

```
for ( ; *s ; s++) cout<<"*";
```

for循环的执行次数是__1__。

参考答案1: 5

设有以下定义：

```
char *s="\ta\017bc";
```

则指针变量s指向的字符串所占的字节数是__1__。

参考答案1: 6

下面程序是从输入的10个字符串中找出最长的那个串。请填空。

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

```
#define N 10
```

```
void main( )
```

```
{
```

```
    char s[N][81], * t;
```

```
    int j;
```

```

    for (j=0; j<N; j++)           //输入N个字符串
        cin>>__1__ ;
    t=*s;           //t指向最长的字符串
    for (j=1; j<N; j++)
        if (strlen(t)<strlen(s[j])) __2__ ;
    cout<<"最长的字符串"<<t<<"的长度"<<strlen(t)<<endl;
}

```

参考答案1: s[j] 或 *(s+j) 2: t=s[j] 或 t=*(s+j)

若已定义:

```
int a=5, b, *p=&b;
```

则语句:

```
*p=a;
```

是把变量a的__1__赋给指针变量p所指变量。

参考答案1: 值

若有说明语句:

```
int c, *d=&c;
```

则通过指针变量d从键盘读入一个整数赋给变量c的语句是:

```
cin>>__1__;
```

参考答案1: *d

变量的指针, 其含义是指该变量的__1__。 参考答案1: 地址

下面程序的功能是将八进制正整数字符串转换为十进制整数。请填空。

```
#include <iostream>
```

```
using namespace std;
```

```

void main( )
{
    char *t, s[8];
    int n;
    __1__ ;
    cin>>t;          //输入八进制数到s中
    n=*t-48;          //48为字符0的ASCII码
    while ( __2__ !=' \0' )    n=n*8+*t-48;
    cout<<n<<endl;
}

```

参考答案1: t=s; 或 t=s 2: *(++t) 或 *++t

下面程序的功能是在字符串s中找出最大的字符并放在第一个位置上，并将该字符前的原字符往后顺序移动，如：boy&girl变成 ybo&girl。请填空。

```

#include<iostream>
using namespace std;
void main( )
{
    char s[80], *t, max, *w;
    t=s;
    cin>>t;
    max=*(t++);
    while ( *t )          //查找最大字符
    {

```



```

        if (max<*t)  {  max=*t;  w=t;  }
        t++;
    }
    t=w;
    while ( __1__ )
    {
        *t=*(t-1);
        __2__ ;
    }
    *t=max;
    cout<<t<<endl;
}

```

参考答案1: $t > s$ 或 $s < t$

2: $t--$ 或 $--t$ 或 $t=t-1$ 或 $t-=1$

有以下函数:

```

char *fun(char *s)
{
    ...
    return s;
}

```

该函数的返回值是形参s所指单元的__1__。

参考答案1: 地址

有如下程序段:

```
char b1[8]="abcdefg", b2[8], *pb=b1+3;
strcpy(b2, pb);
cout<<strlen(b2)<<endl;
```

执行后的输出是__1__。

参考答案1: 4

下面程序统计字符串sub在母串s中出现的次数，其中函数count用于计算q字符串在p字符串中出现的次数。请填空。

```
#include <iostream>
using namespace std;
int count( char *p, char *q)
{
    int m, n, k, num=0;
    for (m=0; p[m]; m++)
        for ( __1__ , k=0; q[k]==p[m]; k++, n++)
            if ( __2__ ]==' \0' )           //测试q所指的下一个字符
            是否为字符串结束符
                { num++; break; }
    return (num);
}
void main()
{
    char s[80], sub[80];
    int n;
    cin>>s;
```

```

        cin>>sub;

        cout<<count(s, sub)<<endl;
    }

```

参考答案1: n=m 2: k+1 或 1+k

已知语句“cout<<s;”的输出是“apple”，则执行语句“cout<<s+2;”的输出结果为__1__。参考答案1: ple

阅读程序，写出程序输出结果：

```

#include <iostream.h>

#include <iomanip>

using namespace std;

int table[5]={5, 4, 3, 2, 7 };

void f (int a[], int n, int *p)
{
    int k, x;
    x=a[0];
    for ( k=1; k<n; k++)
        if ( a[k]<x ) x= a[k];
    *p =x;
}

void main()
{
    int a;
    f(table, 5, &a);
}

```

```
        cout<<a<<endl;
    }
```

参考答案 2

有以下函数：

```
#include<iostream>

using namespace std;

void swap(int *a, int *b)
{
    int *t;
    t=a;    a=b;    b=t;
}

void main()
{
    int x=3, y=5, *p=&x, *q=&y;
    swap(p,q);
    cout<<*p<<*q<<endl;
}
```

写出执行后的输出结果。

参考答案 35

有以下函数：

```
#include<iostream>

using namespace std;

int *p;
```

```

fun( int x, int *y)
{
    int z=4;
    *p=*y+z;
    x=*p-z;
    cout<<x<<*y<<*p<<" ";
}

void main()
{
    int x=1, y=2, z=3;
    p=&y;
    fun(x+z, &y);
    cout<<x<<y<<*p<<endl;
}

```

写出执行后的输出结果。

参考答案 266 166

有如下程序：

```

#include<iostream>
using namespace std;
void fun( int *x, int *y)
{
    cout<<*x<<*y;
    *x=3;
}

```

```

        *y=4;
    }
void main()
{
    int x=1, y=2;
    fun(&y, &x);
    cout<<x<<y<<endl;
}

```

写出执行后的输出结果。

参考答案 2143

有如下程序：

```

#include<iostream>
#include <cstring>
using namespace std;
void fun( char *s )
{
    char t[10];
    s=t;
    strcpy(s, "example");
}
void main()
{
    char t[10]="abcd";
}

```

```
    fun(t);  
    cout<<t<<endl;  
}
```

写出执行后的输出结果。

参考答案 abcd

有以下程序：

```
void ss( char  *s, char  t)  
{  
    while ( *s )  
    {  
        if ( *s==t )  *s=t-' a' +' A' ;  
        s++;  
    }  
}  
  
void main()  
{  
    char  str[100]="cddfdbd", c=' d' ;  
    ss(str, c);  
    cout<<str<<endl;  
}
```

写出程序运行后的输出结果。

参考答案 cDDfDbD

写出以下程序的输出结果：

```

#include<iostream>
using namespace std;
char cchar(char ch)
{
    if (ch>=' A'  && ch<=' Z' )  ch=ch-' A' +' a' ;
    return  ch;
}
void main()
{
    char s[]="AB+bc=deEF", *p=s;
    while ( *p )
    {
        *p=cchar(*p);
        p++;
    }
    cout<<s<<endl;
}

```

参考答案 ab+bc=deef

请写出下面程序运行的结果。

```

#include<iostream>
#include <cstring >
using namespace std;
void main( )

```



```

{
    char s[]="example!", *t;
    t=s;
    while( *t!=' \0' )
    {
        cout<<char(*t-32);
        t++;
    }
}

```

参考答案 EXAM

若有下面的说明和定义，则sizeof(struct aa)的值是()。

```
struct aa
```

```

{
    int r1;
    double r2;
    float r3;
    union uu
    {
        char u1[5];
        long u2[2];
    }ua;
}mya;

```

A 30 B 29 C 24 D 22 参考答案 C

若有定义：

```
struct AA
```

```
{
```

```
    int a;
```

```
    char b;
```

```
    double c;
```

```
}x;
```

则x占用空间大小为__1__字节。

参考答案1： 13

有以下定义和语句：

```
struct date
```

```
{
```

```
    int day;
```

```
    int month;
```

```
    int year;
```

```
    union
```

```
    {
```

```
        int s1;
```

```
        float s2;
```

```
    }share;
```

```
} a;
```

则sizeof(a)的值是__1__。

参考答案1： 16

有下列程序：

```
union b
{
    int k;
    char c[4];
}a;
void main()
{
    a.k=0x3231;
    cout<<a.c[0]<<a.c[1]<<endl;
}
```

其执行结果为__1__。

参考答案1： 12

有以下程序：

```
union pw
{
    int i;
    char ch[2];
} a;
main()
{
    a.ch[0]=13;
    a.ch[1]=0;
```

```
        cout<<a.i<<endl;
    }
```

其输出结果是__1__。

参考答案1: 13

设MClass是一个类，有以下定义：

```
MClass dd,*pp=&dd;
```

则指针变量pp可以通过（ ）访问对象dd的成员。

A :: B . C & D -> 参考答案 D

以下不属于类访问权限的是（ ）。

A public B static C protected D private 参考答案 B

已知f1(int)是类A的公有成员函数，p是指向成员函数f1()的指针，采用（ ）是正确的。

Ap = f1 B p = A::f1 C p = A::f1() D p = f1() 参考答案 B

已知p是一个指向类A的公有数据成员m的指针，A1是类A的一个对象。如果要将A1的m赋值为5，（ ）是正确的。

AA1.p = 5 B A1->p = 5 C A1.*p = 5 D *A1.p = 5 参考答案 C

下列程序中，划线处正确的语句是（ ）。

```
class Base
{
public:
    void fun() { cout<<"Base::fun"<<endl; }
};
class D :public Base
{
```

```

public:
    void fun()
    {
        _____ ;           //显式调用基类的函数fun()
        cout<<"D::fun"<<endl;
    }
};

```

A fun() B Base.fun() C Base::fun() D Base->fun() **参考答案 C**

有类定义如下：

```

class X
{
    int a;
public:
    X( int x=0 ) { a=x; }
};

class Y : public X
{
    int b;
public:
    Y( int x=0, int y=0 ) : X(y) { b=x; }
};

```

下列定义对象的语句组中出现语法错误的是()。

A X *pa=new Y(1,2); B X a1=Y(1,3); C Y b1(2,3); X &a3=b1;

D X a4(10); Y b2=a4; 参考答案 D

控制格式I/O的操作中，（ ）是设置域宽的。

A ws B oct C setfill() D setw() 参考答案 D

下列函数中，（ ）是对文件进行读操作的。

A get() B write() C seekg() D put() 参考答案 A