一．选择填空题

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

A、 “Sjtu” B、 “A Sjtu” C、 “A” D、 ‘A’

2．ch 为 char 类型，判断 ch 为数字字符的表达式是_____B_____

A、‘0’<= ch <= ‘9’ B、(ch >= ‘0’) && (ch <= ‘9’)

C、(ch >= ‘0’) || (ch <= ‘9’) D、(ch >= ‘0’) & (ch <= ‘9’)

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

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

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

4．以下哪个是 C++合法的变量名____D____

A、%store B、12_store_5 C、12_store-5 D、_store_5

5．执行 double x; cin>>x; 问 y 如何能获得 x 四舍五入的结果____A____

A、int y=x+0.5 B、int y=(int)x C、int y=x; D、int y=x-0.5

6．C++语言的跳转语句中，对于 break 和 continue 说法正确的是____C____

A、break 语句只应用于循环体中

B、continue 语句用于跳出循环语句

C、continue 语句用于跳出当前的循环周期

D、break 语句用于跳出当前的循环周期

7．执行语句 int a[10]={3, 5}; 后 a[5]的值为：____D____

A、5 B、3 C、随机值 D、0

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

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

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

9．关于函数的定义，下面哪个说法是错误的____D____

A、一个函数最多只能返回一个值 B、void 类型的函数也可以使用 return 语句

C、函数可以有参数也可以没有 D、函数的返回类型一定和函数的某个参数类型一致

10．在同一工程文件中，一个文件想要共享使用在另外一个文件中声明的一个全局变量，则需要在本文件声明该全局变量前加限定词_____B_____

A、 auto B、 extern C、static D、 register

11．有如下定义语句：int a[] = {1,2,3,4,5};，则对语句 int *p=a;正确的描述是_____B_____

A、语句 int *p = a; 定义不正确

B、语句 int *p=a; 初始化变量 p，使其指向数组对象 a 的首个元素

C、语句 int *p=a; 是把整个数组的值存放在变量 p 中

D、语句 int *p=a; 是把 a[0]的值赋给变量*p

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

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

13．任意一个类，关于构造函数和析构函数，哪个说法是正确的_____A_____【易错】

A、 构造函数可以有多个，析构函数只能有 1 个

B、 构造函数可以有多个，析构函数也能有多个
C、 构造函数只能有 1 个，析构函数也只能有 1 个
D、 构造函数只能有 1 个，析构函数可以有多个

14. 当用 class 来定义一个类时，下面有关类成员的叙述中,不正确的是____C____
A、 当不指定类成员的访问权限时则为私有成员
B、 友元函数可访问类中的任一成员
C、 指定为 public 的成员不允许在类外访问
D、 成员函数可访问类中的任一成员

15. 当通过一个对象来调用参数表为空的静态成员函数时，下面的描述中正确的是_____B_____
A、静态成员函数有 this 指针作为函数的隐含参数
B、静态成员函数不能访问该对象的非静态数据成员
C、静态成员函数不能访问该对象的公有数据成员
D、静态成员函数不能访问该对象的私有数据成员

16. 整数在计算机内部中通常用_____C____来表示。【易错】
A. 原码 B. 反码 C. 补码 D. ASCII 码

17. 已知变量 ch 的类型说明为 char ch; 则以下不符合 C++语法的表达式是____D____【易错】
A、ch='h' B、ch = 32 C、ch = '\a' D、ch ='hi'

18. 下列哪个不是 C++语言中合法的变量名？ ___A_____
A. %correct B. b4hand C. _stk_depth D. MonthTotal

19. 下列代码的输出结果为___A___
```
int a=3, b=5, c=7;
if(a>b)
    a=b-c;
    c=a;
if(c!=a)
    c=b;
cout << a << "," << b << "," << c <<endl;
```
A. 3,5,3 B. 2,5,2 C. 3,6,6 D. 程序中有语法错误

20. 下面哪一表达式不能正确表示数学关系 a<x≤b？ ___D_____
A. a<x&&x<=b B. x<=b&& a<x
C. !(a>=x)&&!(x>b) D. a<x<=b

21. 执行 int a = 1, b = 2, c = 2, d = 4; bool m=1, n=1; bool result=(m = a > b) && (n = c > d);
则 m, n, result 的值分别为___A___【易错】
A. 0, 1, 0 B. 0, 0, 0 C. 1, 2, 0 D. 1, 0, 0

22. 字符串"1234567890"至少需要___B_____字节的存储空间
A. 10 B. 11 C. 40 D. 44

23. 以下哪个选项正确分配了大小为 10 个整数的动态数组？ ___B____
A. int *a = new int(10);
B. int *a = new int[10];
C. int *a = new int(10*sizeof(int));
D. int *a = new int[10*sizeof(int)];

24. 以下所列的各函数原型中，正确的是____C____
A. (int, double) play(int a, int b);　　B. double play(int a=10, int b);
C. void play(int *a, int b);　　D. int play(int a, b);

25. 以下正确的说法为____D____
A．任何函数都不可以调用自己　　B．函数可以有多个返回值
C．一个程序可以有多个 main 函数　　D．函数的形式参数是其局部变量

26. 假定有语句“ int b[10], *pb; ”，则不正确的赋值为___B___
A. pb=b　　B. pb=b[5]　　C. pb=b+2　　D. pb=&b[0]

27. 已知 int a[]={1,2,5,3} 下列哪一个表达式的结果为 3？___C___【易错】
A. *(&a+2)　　B. *(a+2)　　C. *a+2　　D. *(&a[0]+2)

28. 当一个程序中调用库中的函数时，以下说法正确的是 ___A___【易错】
A. 需 include 该库的头文件
B. 除了该库的头文件，不需要其它与该库相关的任何文件
C. 直接调用库函数就可以，程序中不需做任何与该库相关的操作
D. 一定要有库函数实现的源码文件

29. 关于构造函数和析构函数的定义，下面正确的选项是____C____。
A. void X::X(参数)，void X:: ~X(参数)　　B. X::X(参数)，X:: ~X(参数)
C. X::X(参数)，X:: ~X( )　　D. X::X( )，X::~X(参数)

30. 下面对于友元函数描述正确的是____D____。
A. 友元函数不能访问类的私有成员
B. 友元函数的实现必须在类的内部定义
C. 友元函数是类的成员
D. 友元函数破坏了类的封装性和隐藏性

31. 设有语句 char ch; cin.get(ch); 下列能够输出 ch 的 ASCII 码的语句是__A___。
A. cout<<(int)ch;　　B. cout<<”ch”;　　C. cout<<ch;　　D. cout<<ch+’0’;

32. 有如下的枚举定义：
enum color { Monday=1, Tuesday, Wednesday, Thursday, Friday, Saturday=11, Sunday};
则 Wednesday 和 Sunday 的值分别为___C_____。
A. 3, 7　　B. 2, 6　　C. 3, 12　　D. 2, 7

33. 执行下列语句后，s 的值是多少___D_____。
```
int s=0;
for(int i=0;i<=10;i++)
   if(i%2==0 || i%3==0) continue; else s+=i;
```
A.12　　B.11　　C.14　　D.13

34. 有整型变量 y=5; 运行代码的结果是显示: NOT OK. 以下正确的代码块是__C____。

A.
```
y=(y>1)?0:10;
if (y)
cout<<"NOT";
cout<<" OK";
```

B.
```
if (10>y>2)
cout<<"NOT";
if(2<y<10)
cout<<" OK";
```

C.
```
switch(y){
   case 5:
   cout<<"NOT";
   default :
   cout<<" OK";
}
```

D.
```
while(y>=5)
{
  cout<<"NOT";
}
cout<<" OK";
```

35. 下列循环语句有错误的是___A_____。
A. int i = 0, sz = 10; for (i != sz; ++i) {…}　　B. for (int i=0, sz=10; i < sz; ++i) {…}
C. while (1) {…}　　D. int i = 0, sz = 1; while (i != sz) {…}

36. 以下对跳转语句 continue 和 break 的叙述错误的是___D____。
A. continue 语句只能出现在循环内部　　B. break 语句可以用于 switch 语句中
C. continue 语句终止最近的循环中的当前迭代，并立即开始下一次迭代
D. 若 while 循环中包含 switch 语句，switch 语句中的 break 可以跳出 while 循环

37. 执行 C++语句 cin.getline(str, 80, '$'); 从输入流读取字符串后，字符数组 str 中不可能包含的字符是__C__。
A. '\n'　　B. ' '　　C. '$'　　D. '\0'

38. 已知 char str[10] = "2019\t"；则 strlen(str)+sizeof(str)的值是__B__ 。【易错】
A. 小于等于 14　　B. 15　　C. 16　　D. 17

39. 以下有关函数的说法，正确的是__B____ 。【易错】
A. 不能定义两个函数名一样的函数　　B. 两个函数的原型声明不能相同
C. 函数返回时所有局部变量都会自动被撤销
D. 参数传递是值传递，数组作为参数传递时，数组按元素逐一传递

40. 下面有关变量生命周期的说法，正确的是___C__ 。
A. 自动变量是由系统自动在堆区域为其分配空间，当定义它的函数返回时自动消亡
B. 静态局部变量存在于系统的栈区域，会随着定义的函数返回而消亡
C. 静态局部变量和静态全局变量一样，都是在整个程序退出时才消亡
D. 以上说法都是错误的

41. 已知函数 fun 的原型是：void fun(double *a, char &b,); 变量 v1、v2 的定义是：
double v1[100]; char v2;，正确的调用语句是___B____ 。
A. fun (v1, &v2) ; B. fun (v1, v2); C. fun (&v1, &v2) ; D. fun (&v1, v2);

42. 若有如下定义，则对数组元素的成员引用不正确的是____C___ 。【易错】
struct Student { int id; char name[20]; char gender; int age; };
Student stu [2], *p; p=stu;
A. p[1].gender B. p->age　　C. p[0]->id　　D. stu[1].name

43. 当说明一个结构体变量时，系统分配给它的内存是_____A_____。
A. 结构体各成员所需内存量的总和　　B. 结构体中第一个成员所需内存量
C. 结构体成员中占内存量最大者所需内存量　　D. 结构体中最后一个成员所需内存量

44. 能够释放对象所占资源的是____A_____。
A. 析构函数　　B. 拷贝构造函数　　C. 构造函数　　D. 静态成员函数

45. 已知 A 类，则当程序执行到语句：A a[4],*pa[3];时，调用了___B____次构造函数。【易错】
A. 3　　B. 4　　C. 7　　D. 8

46. 下列说法中错误的是 ____B____。
A. 公有继承时基类中的 public 成员在派生类中仍是 public 的
B. 公有继承时基类中的 private 成员在派生类中仍是 private 的
C. 私有继承时基类中的 public 成员在派生类中是 private 的
D. 保护继承时基类中的 public 成员在派生类中是 protected 的

47．执行以下语句： char ch[10], ch1[10]; cin.getline(ch, 8, '.'); cin.getline(ch1, 8, '.');
当用户输入：abc.def.（回车）后 cout<<ch1 的结果为：____A____。
A. “def” B. “.def” C. “def.” D. “abc.def.”

48．设有整型数组 a[10]，要输出数组 a 的 10 个元素，错误的是：____B____【易错】
A. for (int *p=a, int i=0; i<10; ++i) cout << p[i];
B. for (int i=0; i<10; ++i) { cout << *a; ++a; }
C. for (int *p=a; p<a+10; ++p) cout << *p;
D. for (int i=0; i<10; ++i) cout << *(a+i);

49．以下关于函数参数的叙述不正确的是：____B____【易错】
A. 函数的参数不仅仅可以是函数的输入，还可以是函数的输出
B. 对象作为实参时系统将自动调用拷贝构造函数
C. 函数的形参命名可以任意，只要符合标识符规则
D. 函数参数的值可以是内存单元地址

50．已知程序
```
#include <iostream>
using namespace std;
int main()
{ for(int i=5; i>1; --i) cout <<i; cout << i; return 0; }
```
则运行该程序之后，输出：____E____【易错】
A. 5432 B. 54321 C. 55443322 D. 543210 E. 以上都不对

51．已知语句 myfun(x) = yourfun(y);可以正常运行，以下说法中正确的是：____C____【易错】
A. myfun 函数的参数只能采用引用传递
B. yourfun 函数的参数只能采用引用传递
C. myfun 函数不能返回非静态的局部变量
D. yourfun 函数不能返回非静态的局部变量

52．以下程序段中，可以运行并且得到合理正确的结果的是：____D____【易错】
A. char s[]={‘1’,’2’,’3’}; cout<<s;
B. int x=12345678; x*=x; cout << x;
C. int n; cin>>n; if(n=0) cout << “It is Zero”; else cout << “It is not zero”;
D. 以上选项都不能得到合理正确的结果

53．short int 类型的-1 在内存中存储为_____C_____【易错】
A. 0x0001 B. 0xFFFE C. 0xFFFF D. 0x8001

54．下面不合法的字符常量是_____D_____【易错】
A. '\\' B. '\012' C. '\0' D. "a"

55．已知 int 型整数所能表示的数值范围为[INT_MIN,INT_MAX]，为避免 int 型数 x 和 y 相加溢出，以下代码正确的是_____所有选项都错_____
A. if(x+y>INT_MAX) cout<<”error” else cout<<x+y
B. if(x+y>INT_MAX && x+y<INT_MIN) cout<<”error” else cout<<x+y
C. if(x>INT_MAX-y && x<INT_MIN-y) cout<<”error” else cout<<x+y
D. if(x+y>INT_MAX || x+y<INT_MIN) cout<<”error” else cout<<x+y
E. if(x>INT_MAX-y || x<INT_MIN-y) cout<<”error” else cout<<x+y
F. 以上表达式至少有 2 个是正确的

56．执行完以下语句后，count 的值的表达式是_____A_____

for(count=0, i=0; i<n; i++) for(j=0; j<i; j++, count+=2);
A. n*n-n　　B. n*n　　C. n*n-1　　D. 以上都不对

57. 以下每组有 4 个运算符，其中优先级从左向右逐次降低的是____E____
A. / + == && 　　B. ! % <= = 　　C. () % || =
D. -> / > , 　　E. 以上都正确

58. 若正整数 x 使得 x%5==0 && !(x%7)为真，则 x ____A_____
A. 既是 5 的倍数又是 7 的倍数　　B. 是 5 的倍数但不是 7 的倍数
C. 不是 5 的倍数但却是 7 的倍数　　D. 既不是 5 的倍数又不是 7 的倍数

59. 若在执行了语句"double a[10]={3.8, 2.9};"后，执行 "cout<<a[3];"，那么输出的值为____C____
A. 3.8　　B. 2.9　　C. 0　　D. 随机值

60. 下面的 do while 语句的循环体执行次数为____D____【易错】
```
unsigned int a = 10;
do {
    a--;
} while(a>=0);
```
A. 9　　B. 10　　C.11　　D. 无数次

61. 如下函数模板中的 T 是____C____
```
template <class T> T square(T x) { return x * x; }
```
A. 函数形参　　B. 函数实参　　C. 模板参数　　D. 模板实参

62. 对于如下函数声明为，则下面调用写法正确的是____B____【易错】
```
void fun(int a, int b=1, char c='a', double d=3.2);
```
A. fun();　　B. fun(2,3);　　C. fun(2, ,'c', 3.14);　　D. fun(int a=1);

63. 有以下程序段：
```
struct studentT {
    char    no[10];
    char    name[10];
    int     chinese;
    int     math;
    int     english;
};
int main() {
    studentT    student1,    *sp;
    _____________________
    //下略
}
```
那么空白处可以填入的语句是____B____
A. student1->chinese=90;　　B. student1.chinese=90;
C. (*sp).chinese=90;　　D. sp->chinese=90;　　E. sp.chinese=90

64. 以下有关类的静态成员的描述正确的是____A____
A. 在建立对象前，就可以为静态数据成员赋值
B. 静态成员函数在类外定义必需要用 static 前缀
C. 静态成员函数只能在类外定义
D. 在静态成员函数中可以使用 this 指针

65. 下列的各类函数中，哪一个不是类的成员函数____B____

A. 构造函数　　B. 友元函数　　C. 析构函数　　D. 拷贝构造函数

66. 已知类 A 是类 B 的友元，类 B 是类 C 的友元，则正确的描述是____B____
A. 类 A 一定是类 C 的友元　　B. 类 A 的成员函数可以访问类 B 的对象的私有成员
C. 类 C 一定是类 A 的友元　　D. 类 C 的成员函数可以访问类 B 的对象的私有成员

67. 假设父类 A 中某成员 a 的访问特性为 protected，从父类 A public 派生出子类 B，则在子类 B 中对成员 a 的访问特性为____B____
A. public　　B. protected　　C. private　　D. 不可访问

68. 以下说法正确的是___A____
A. 指针不仅包含地址信息，还包含类型信息
B. 不同类型的指针所占内存空间大小也不同
C. 通过指针访问的是属于堆的内存空间
D. 有定义 int*p，如果 p 的值为 1000，则 p+1 的值为 1001

69. 语句___B____可以输出字符型变量 ch 中存储的字符的 ASCII 值
A. cout<<ch;　　B. cout<<(int)ch;　　C. cout<<(asc)ch;　　D.cout<<&ch;

70. ___B____是 double 常量【易错】
A. 01234　　B. 1e2　　C. 0xee　　D. 以上都不是

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

72. 一个源文件想要定义一个本源文件专用、不允许其它源文件共享的全局变量，则需要在本文件定义该全局变量前加限定词___C____
A. auto　　B. extern　　C. static　　D. register

73. 若有函数 f(int*a, const int &b, int &c)以及定义 int m;和 const int n=9:，以下合法的函数调用是____B___【易错】
A. f(&m, 5, 5)　　B. f(&m, 5, m)　　C. f(&n, n, n)　　D. f(m, m, n)

74. 以下关于函数的描述中说法正确的是___B____【易错】
A. 函数调用 f(1, , 'x')表示 f 的第二个参数将使用默认值
B. 处理逻辑完全相同而被处理的数据类型不同的多个函数可以写成一个函数模板
C. 重载函数是一组名字相同的函数，但是这些函数或者参数个数不同，或者参数个数相同而类型不同，或者参数完全相同但是返回类型不同
D. 以上至少有两个叙述正确

75. 在函数声明时，下列____D___项是不必要的。
A. 函数的返回类型　　B. 函数形参类型　　C. 函数的名字　　D. 形参的名字

76. 要将一个二维数组 int a[3][4]传递给函数 f，以下函数原型____A___是正确的【易错】
A. void f(int[][4], int);　　B. void f(int[3][4]);
C. void f(int *, int);　　D. void f(int **, int);

77. 以下有关数组 int a[10]与指针 int *p 的说法正确的是___A____

A. 数组名 a 表示的是整个数组的起始地址
B. 数组名 a 表示的是数组首元素的地址，且是指针常量，因此表达式 a+1 是不合法的
C. 逗号表达式 p=a, p[10]=-5 中的两个赋值都是合法的
D. 在 32 位的计算机中，sizeof(a)和 sizeof(p)的结果都是 4

78. 以下哪些函数能够用于两个数的交换? D 【易错】

void swap1(int p, int q)
{
int temp,
temp=p;
p=q:
q=temp;
}
	void swap2(int *p, int *g)
	{
	int *temp;
	*temp=*p;
	*p=*q;
	*q=*temp;
	}
		void swap3(int *p, int *q)
		{
		int *temp;
		temp=p;
		p=q;
		q=temp;
		}
			void swap4(int &p, int &q)
			{
			int temp;
			temp=p;
			p=q;
			q=temp;
			}

A. 只有 swap1　B. 只有 swap2　C. 只有 swap3　D. 只有 swap4
E. 有 2 个可以　F. 有 3 个可以　G. 都可以　H. 都不可以

79. 已知有如下结构体定义

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

在单向链表中，如何将一个新节点 newNode 插入到非空节点 current 的后面? A

A. newNode->next = current->next; current->next = newNode;
B. current->next = newNode; newNode->next = current->next;
C. current->next = newNode;
D. newNode->next = current;

80. 派生类的构造函数的初始化列表中，不能包含 C
A. 基类的构造函数　B. 派生类中成员对象的初始化
C. 基类中成员对象的初始化　D. 派生类中一般数据成员的初始化

81. 如果一个 C++类的成员被声明为 static，则肯定不正确的是 A
A. 如果是成员函数，该函数可以访问任何数据成员
B. 不论是数据成员还是成员函数，都不需要类的实例就可以使用
C. 如果是数据成员，任意成员函数都可以对其访问
D. 如果是数据成员，可以视为类内的全局变量

82. 以下关于构造函数的 5 项描述中，不正确的个数是 D 【易错】
1) 对于函数声明 void f(A &);当用 A 的对象 a 调用 f(a)时会自动调用拷贝构造函数
2) 拷贝构造函数的参数通常采用引用传递，但是也能够使用值传递，只是效率低一些
3) 如果有类 A，则语句 A a[4], *p; 会调用构造函数 4 次
4) 在对象之间互相赋值时会自动调用拷贝构造函数
5) 如果程序员未创建，系统可以自动生成一个无参数的构造函数和一个拷贝构造函数
A. 0　B. 1　C. 2　D. 3　E. 4　F. 5

二．给出下列程序段的运行结果：

1.

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

```
    char str[30];
    cin>>str;
    for(int i = 0; str[i]!='\0';++i){
        if(str[i]>='A'&&str[i]<='Z')    str[i] -= 1;
        if(str[i]>='a'&&str[i]<='z')     str[i] += 2;
        if(str[i]>'0'&&str[i]<='9')      str[i] -= 1;
    }
    cout<<str<<endl;
    return 0;
}
```

当从键盘输入字符串”Cwc3029!”时，程序的输出结果:

答案：Bye2018！

2.

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

答案：30 31 30 31 30

3.

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

答案：*p[0]=C（换行）*p[1]=_ （换行）*p[2]=o（换行）*p[3]=a（换行）*p[4]=i

4. 写出调用 foo(6)的运行结果

```
int foo(int n)
{    int temp;
    switch(n){
      case 0:
```

```
        case 1: return 1;
        default:    temp= (n-1)*foo(n-2);
                    cout<<temp<<" ";
                    return temp;
    }
}
```
答案：1 3 15（如果将“调用 foo(6)”理解为“cout<<foo(6);”，那么 1 3 15 15 也算对）

5.
```
#include <iostream>
using namespace std;
void fun(int x, int *y, int &z)
{    x=*y+z;
     *y=x+z;
     z=x+*y;
}
int main()
{    int a=1, b=2, c=3;
     fun(a, &b, c);
     cout<<a<<" "<<b<<" "<<c<<endl;
}
```
答案：1 8 13

6.
```
#include <iostream>
using namespace std;
int swap(int &x, int *y, int z=1)
{    int *a = &x;
      static int b=x;
      b+=2;
      *a=(*a)-z+b;
      return b;    }

int main()
{    int a=8, b=5;
      cout<<swap(a, &b)<<" "<<a<<" "<<b<<endl;
      cout<<swap(a, &b)<<" "<<a<<" "<<b<<endl;
      return 0;
}
```
答案：10 17 5（换行）12 28 5

7.
```
#include <iostream>
using namespace std;
class Student {
      friend void fun (Student &rs);
      private:
            int num;
            double score;
            static int count;
      public:
            Student (int i=0, double j=0)
                  { num=i; score=j; count++;}
            void Print ()
```

```
        {cout<< num<<','<<score<<endl;}
    static void disp(Student &ss) {cout<<"ss.count:"<<ss.count<< " count:"<<count<<endl;}
};
int Student::count = 0;
void fun (Student &rs) {
    rs.num=2018;
    rs.score=90;
    rs.Print ();
}
int main ()
{
    Student s1 (1008,60), s2;
    s1.Print ();
    fun (s2);
    Student::disp(s1);
    return 0;
}
```

答案：1008, 60（换行）2018，90（换行）ss.count:2 count:2

8.

```
#include <iostream>
using namespace std;
int main(){
    int n=21;
    cout<<n;
    while(n!=1){
        if(n%2==0){
            n/=2;
            cout<<"->"<<n;
        }
        else{
            n=n*3+1;
            cout<<"->"<<n;
        }
    }
    return 0;
}
```

答案：21->64->32->16->8->4->2->1

9.

```
#include <iostream>
using namespace std;
int main(){
    for (int i=0; i < 3; ++i){
        int j = 5;
        while(i < j){
            j = j - 3 - i;
            cout << '*';
        }
        switch (j) {
        case -1:
            cout << 'A';
            break;
        case 0:
```

```
                cout << 'B';
                break;
        case 1:
                cout << 'C';
        default:
                cout << 'E';
        }
    }
    return 0;
}
```

答案：**A*CE*B

10.

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

答案：4 9 16 25 1

11.

```
#include <iostream>
using namespace std;
class CC
{
private:
    int A,B;
public:
    CC(int i,int j)
    {A=i;B=j;cout<<"Call constructor\n";}
    CC(const CC &obj)
    {A=obj.A+1;B=obj.B+2;cout<<"Call copy constructor\n";}
    ~CC()
    {cout<<"Call destructor\n";}
    void print()
    {cout<<"A="<<A<<", B="<<B<<endl; }
};
int main()
{   CC a1(2,3);
    CC a2(a1);
    a2.print();
```

答案:
Call constructor
Call copy constructor
A=3, B=5
Call constructor
A=5, B=6
Call destructor
Call destructor
Call destructor

```
    CC *p = new CC(5,6);
    p->print();
    delete p;
    return 0;
}
```

12.

```
#include <iostream>
using namespace std;
int main()
{
    int a[5] ={1,3,5,9,2};
    for (int i=0; i<5; i++)
        switch (a[i]+1)
        {   case 4:
                cout<<"first ";
                break;
            case 2:
                cout<<"second ";
                break;
            case 10:
                cout<<"third ";
            default:
                cout<<"forth ";
                cout<<endl;
        }
    return 0;
}
```

答案：second first forth（换行）third forth（换行）forth

13.

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

答案：16, 28（换行）16, 29

14.

```
#include <iostream>
using namespace std;
int main( ){
	char a[3] = {'c','a','g'};
	int b[3] = { 'f','a','p'};
	int c[3] = { 0 };
	int j;
	for (int i = 0; i<3; i++){
		j = 0;
		while ((j<3)&&(a[i] <= b[j]))
		{ j++; c[i]++; }
	}
	for (int i= 0; i<3; i++)
		cout << c[i] << " ";
	cout << endl;
	return 0;
}
```

答案：1 3 0

15.

```
#include <iostream>
using namespace std;
int main()
{
		int s = 0;
		int i;
		for (i = 1;; i=i+2) {
			if (s>10)
				break;
			if (i % 3 == 0)
				s += i;
		}
		cout << "i,s=" << i << "," << s << endl;
		return 0;
}
```

答案：i,s=11,12

16.

```
#include <iostream>
using namespace std;
class Tdate
{
public:
	void set(int,int,int);
	int isLeapYear();
	void print();
private:
	int month;    int day; int year;
};

void Tdate::set(int m,int d,int y=2000)
{	 month=m;day=d;year=y;	}
```

```
int Tdate::isLeapYear()
{       return((year%4==0&&year%100!=0)||(year%400==0));    }
void Tdate::print()
{
      cout<<year<<"/"<<month<<"/"<<day<<endl;
}
int main()
{
   Tdate s, *pTdate=&s;
   s.set(12,2,2016);
   pTdate->print();
   if((*pTdate).isLeapYear())
      cout<<"error"<<endl;
   else
      cout<<"right"<<endl;
   return 0;
}
```

答案：2016/12/2（换行）error

17.

```
#include <iostream>
#define N 25
using namespace std;
int main()
{
      int x = N;
      int y;
      switch (x/10) {
            case 0 : y = x; break ;
            case 1 : y = x/2+1.5; break ;
            case 2 : y = x/2+2.5;
            case 3 : y = x/2+3.5; break ;
            default : y = x/2;
      }
      cout<<y<<endl;
      return 0;
}
```

答案：15

18.

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

答案：2 -1 0（换行）1 0 1（换行）0 0 0（换行）-1 0 1（换行）-2 -1 0

19.

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

```
void func(int x, int &y)
{
	static int a = 3;
	int z = 2*x;
	x = y-3;
	a = x+y+z;
	y = a+z+10;
}
int main()
{
	int x=2, y=1;
	func(x, y);
	cout << x << ' ' << y << endl;
	func(x+2, y);
	cout << x << ' ' << y << endl;
	return 0;
}
```

答案：2 17（换行）2 57

20.

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

答案：two（换行）wo

21.

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

答案：A（换行）X（换行）B（换行）B

22.

```
#include <iostream>
#include <cstring>
using namespace std;
class test
{
    friend void print(const test &obj) { cout << obj.data << endl; }
    int len;
    char *data;
  public:
    test(const char *s1= NULL) {
        len = strlen(s1);
        data = new char[len+1];
        strcpy(data, s1);
    }
    test(const test &s1) {
        if (s1.len <= 2) {
            len = s1.len;
            data = new char[len+1];
            strcpy(data, s1.data);
        }else {
            len = s1.len - 2;
            data = new char[len+1];
            strcpy(data, s1.data+2);
        }
    }
    ~test() { delete []data; cout << "delete" << endl; }
};
int main() {
    test a("abcde"), b = a;
    test &c = a;
    print(a);
    print(b);
    print(c);
    return 0;
}
```

答案：abcde（换行）cde（换行）abcde（换行）delete（换行）delete

23.

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

答案：10

24.

```
#include <iostream>
using namespace std;
int main( )
{
    int final_score=88;
    char grade;
    switch (final_score / 10)
    {
        case 7: cout << “C” << endl; grade = 'C'
        case 8: cout << “B” << endl; grade = 'B';
        case 9: cout << “A” << endl; grade ='A'; break
        default: cout << “D” << endl; grade = 'D';
    }
    if ( ( ( grade>='A' ) && ( grade <= 'C' ) ) || ( final_score++) )
    {
        cout << final_score << ‘ ’ << grade << endl;
    }
    return 0;
}
```

答案：B（换行）A（换行）88 A

25.

```
#include<iostream>
using namespace std;
int main()
{
    int a[8]={22,6,78,23,56,10,11,20}, n = 8;
    int I, j, k;
    i= a[n - 3];
    j= a[n -2];
    k = a[n - 1];
    for (int i= n- 4; i>= 0; i--)
        a[i + 3] = ali];
    a[0] = i;
    a[1] = j;
    a[2] = k;
    for ( int i= 0; i< n; i++)
        cout << a[i] << " ",
    return 0;
}
```

答案：10 11 20 22 6 78 23 56

26.

```
#include<iostream>
using namespace std;
void f1(int n)
{
    static int m=n;
    if (n==0) return;
    f1(n-1);
    for (int i=0; i<m-n; i++) cout<<' ';
    for (int i=0; i<2*n-1; i++) cout<<'*'
    cout<<endl;
}
```

```
void f2(int n)
{
	static int m=0;
	if(n==0) return;
	m++;
	f2(n-1);
	m--
	for(int i=0; i<m; i++) cout<<" ";
	for(int i=0; i<2*n-1; i++) cout<<"*";
	cout<<endl;
}
int main()
{
	for(int i=3; i>1; i--) f1(i);
	for(int i=3; i>1; i--) f2(i);
	return 0;
}
```

答案:

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

27.

```
#include <iostream>
using namespace std;
int main() {
	int arr[5]= {12, 7, 5, 13, 8};
	int *ptr1 = arr;
	int *ptr2 = arr + 4;
	int ans[5] = {0};
	for (inti = 0; i< 5; i++) {
		int count = 0;
		while (ptr1 < ptr2) {
			if (*ptr1 < *ptr2) {
				count++;
			}
			ptr1++;
		}
		ans{i] = count;
		ptr1 = arr;
		ptr2--;
	}
	for (int i = 0; i< 5: i++)
		cout << ans[i] << “ “;
	cout << endl;
	return 0;
}
```

答案：2 3 0 0 0

28.

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
	ofstream out("data.txt");
	int i;
```

```
    for (i = 1; i <= 10; ++i) out << i << ‘ ’ ;
    out.close();
    fstream in("data.txt");
    in.seekp(10);
    in << 20;
    in.seekg(0);
    do
    {
        in>>i;
        cout<< i << ' ';
    }}while (i<9);
    in.close();
    return 0;
}
```

答案：1 2 3 4 5 207

29.

```
#include<iostream>
using namespace std;
class BASE
{
    char c;
public:
    BASE(char n) : c(n) {}
    void show() { cout << c; }
    virtual ~BASE() { cout << c; }
};
class DERIVED : public BASE
{
    char c;
public:
    DERIVED(char n) : BASE(n + 1), c(n) { }
    void show() { cout << c; };
    ~DERIVED() { cout << c; }
};
int main()
{
    DERIVED d=('X’);
    BASE &b1=d;
    BASE *b2=new DERIVED('Y');
    d.show();
    b1.show();
    b2->show();
    delete b2;
    return 0;
}
```

答案：XYZYZXY

三．程序填空题：

1. 下面函数将一个整数字符串转换为一个整数。

```
    int fun(char* str)
    {
```

```
    int num, digital;

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

2. 下列程序将两个字符串拼接起来。

```
#include <iostream>
using namespace std;
void myStrCat(char *str1,char *str2);
int main(void)
{
    char dst[100] = "Hello, ";
    char src[100] = "Welcome to C++!";
    ____myStrCat(dst, src)____          //调用myStrCat函数
    cout<<dst<<endl;
    return 0;
}
void myStrCat(char *str1,char *str2)
{
    int i=0, len;
    len = strlen(str1);
    while(_____str2[i] != '\0'_____)
    {
        ____str1[len]=str2[i]____       //通过赋值，实现str2字符拼接至str1后面
        len++;
        i++;
    }
    str1[len] = '\0';
}
```

3. 数组中的峰值元素是指其值大于左右相邻元素值的元素。一个数组中可能有多个峰值，如数组元素 [1, 2, 1, 3, 4, 1]中 2 和 4 都是峰值。编程实现数组中所有峰值元素的值。

```
#include <iostream>
using namespace std;
int main()
{   int *arr;
    int cur, num;

    cout << "输入元素个数：";   cin >> num;
    __arr = new int[num];___

    for ( int i = 0; i < num; ++i)
        cin >> arr[i];
    cur = 1;

    while (_____cur < num-1_____) {
        if ((arr[cur-1] < arr[cur])&&(arr[cur+1] < arr[cur]))
            cout << arr[cur] << endl;
```

```
            cur++;
        }

        ____delete [] arr;____
        return 0;
    }
```

4. 填写以下程序中的空格。

```
    void fun(int a[], int n)
    {
        int maxOrmin, pm, t;
        for (int i = 0; i<n - 1; i++)
        {     maxOrmin =  _______a[i]_______  ;
            pm = i;
            for (int j = i + 1; j<n; j++)
            {
                if (((i%2==0)&&(___a[j]<maxOrmin___)) || ((____i%2==1____)&&(a[j]>maxOrmin)))
                {     maxOrmin = a[j];
                    pm = j;
                }
            }
            if (pm!=i)
            {
                t=a[i];
                a[i]=maxOrmin;
                ____a[pm]____=t;
            }
        }
    }
```

5. 填写以下程序中的空格。

```
    #include <iostream>
    using namespace std;
    void getResult(_______int a[], int n, int &max, int &min, double *p_ave__________)
    {
        _double total = 0;_______
        max = min = a[0];
        for (int i=0; i<n; i++)
        {
            if (a[i]>max) max = a[i];
            if (a[i]<min) min = a[i];
            total += a[i];
        }
        *p_ave =total/n;
    }
    int main()
    {
        int a[5];
        int max, min;
        double ave;
        for (int i=0; i<5; i++)
            cin>>a[i];
        getResult(a, 5, max, min,  _&ave__);
        cout<<ave<<" "<<max<<" "<<min<<endl;
```

```
    return 0;
}
```

6. 新鸡兔同笼问题：想数一数自己养了多少动物，输入 1 代表鸡，输入 2 代表兔子，统计一下共有多少只动物，多少条腿。

```
#include <iostream>
using namespace std;
int main()
{
    int sum=0, leg=0, type;
    while(cin>>type)
    {
        sum__++_____________________________;
        leg___+=(type==1)?2:4_____________________;
    }
    cout << "动物总数：" << sum << endl;
    cout << "共有" << leg << "条腿" << endl;
    return 0;
}
```

7. 输出 10000 以内能被 7 整除且个位数为 3 的所有整数，并且使得计算量尽量少。

```
#include <iostream>
using namespace std;
int main()
{
    int   i, j;
    for (i=0; _________i<1000___________; i++)
    {
        j=i*10+3;
        if (_________j%7==0_____________)
            cout << j << " ";
    }
    cout << endl;
    return 0;
}
```

8. 有 n 个整数，使前面 n-m 个数顺序向后移 m(m<n)个位置，最后 m 个数变成最前面 m 个数，如{1， 2，3，4，5}移动 2 个位置得到{4，5，1，2，3}，以下程序使用递归方法实现。

```
#include<iostream>
using namespace std;
class Data
{
    friend void move(const Data &, int )_____________________;
    int *arr, num;
  public:
    Data();
    void printall();
    ~Data(){ __delete [] arr _____________________________________; }
};
void move(const Data &data, int m)
{
    int arr_end,*p;
    arr_end=data.arr[data.num-1];
    for(p=__data.arr+data.num-1____; p>__data.arr_____________; p--)
```

```
        __*p=*(p-1)_____________________________________________;
    data.arr[0]=arr_end;
    m--;
    if(m>0) move(data, m);
}

Data::Data()
{
    cin>>num;
    arr=new int[num]_____________________________________________;
    for (int i=0;i<num; i++)
        cin>>arr[i] ;
}
void Data::printall()
{
    for (int i=0;i<num;i++)
        cout<<arr[i]<<' ';
}
int main( )
{
    int m;
    Data data;
    cout<<"Please enter move number: ";
    cin>>m;
    move(data,m);
    cout<<"After move: ";
    data.printall();
    return 0;
}
```

9. 本程序包含 5 个文件，在同一个目录下，输出为 81/256。本题有 4 个空

```
// main.cpp
#include "qpow.h"
#include "Rational.h"
int main()
{
    Rational a(3, 4);
    qpow(a, 4).print();
    return 0;
}

// qpow.h
#ifndef QPOW_H
#define QPOW_H
#include "Rational.h"_____________________
Rational qpow(const Rational &x, int n); //计算有理数 x 的 n 次幂
#endif

// qpow.cpp
#include "qpow.h"
Rational qpow(const Rational &x, int n)
{
    Rational res(1,1)_____
    for (int i = 0; i< n; ++i)
```

```
            res = res * x;
        return res;
}

// Rational.h
#ifndef RAT_H
#define RAT_H
class Rational { //有理数类，不考虑化简
        int num; //分子
        int den; //分母
public:
        Rational(int n, int d) : num(n),. den(d) { }
        Rational operator *(const Rational &);
        void print();
};
#endif

// Rational.cpp
#include "Rational.h"
#include <iostream>
using namespace std;
Rational Rational::operator*(const Rational &o)
{
        return Rational(num * o.num, den * o.den);
}
void Rational::print()
{
        cout << num << '/' << den;
}
```

10. 字符串数组输入并排序(升序)。本题有 7 个空

```
#include<iostream>
#include <cstring>
using namespace std;
void sortStr(__char **str, int n_____) //对字符串进行排序
{
        char *tmp;
        for(int i=0; i<n; i++)
                for(int j=i+1; j<n; j++)
                {
                        if(____strcmp(str[i], str[j])>0_____)
                        {
                                tmp=str[i]_____;
                                str[i]=str[j]_____;
                                str[j]=tmp_____;
                        }
                }
}
int main()
{
        char *str[30], temp[80];
        int i;
        const int n=30;
        for(i=0; i<n; i++)
```

```
        {
            cin.getline(temp, 80);
            str[i]=new char[strlen(temp)+1];
            strcpy(str[i], temp)_____;
        }
        sortStr(str, n);
        for(i=0; i<n; i++) cout<<str[i]<<endl;
        for(i=0; i<n; i++) delete [] str[i]_____;
        return 0;
    }
```

11. 实现链表查找。本题有 4 个空

```
    #include <iostream>
    using namespace std;
    struct Node {
        int data;
        Node* next;
    };
    Node* findNode(___Node* head, int target_____)
        while(_____head != NULL_____) {
            if(__head->data__== target) {
                return head;
            }
            head=__head->next_____;
        }
        return NULL;
    }
    int main() {
        //建立具有 3 个节点的链表
        Node* head = new Node {1, NULL};
        Node* second = new Node {2, NULL);
        Node* third = new Node {3, NULL};
        head->next = second;
        second->next = third;
        //设定查找目标
        int target = 2;
        //通过函数调用进行查找
        Node* result = findNode(head, target);
        if (result != NULL) cout << "Node with value " << target << " found!” << endl;
        else cout << "Node with value " << target << " not found." << endl;
        delete head;
        delete second;
        delete third;
        return 0;
    }
```

四、编程题

1、输入一个字符串 str，再输入一个字符 c，试完成函数 void delChar (char str[], char c)的实现部分，该函数可将 str 中的字符 c 全部删除。 要求该函数用递归的方法来实现。

答案略

2. 定义一个日期类 TDate，使得下面测试程序

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

        return 0;
}
```

运行结果为：

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

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

答案略

3. 设计一个函数 double Sin(double x, double epsilon)，使用以下无穷级数计算 sinx 的值：

$$sinx = x/(1!) - x^3/(3!) + x^5/(5!) - x^7/(7!) + \ldots$$

要求（1）尽量减少计算量，（2）当 xn/(n!)的绝对值小于事先指定的 epsilon 时结束计算。

答案略

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

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

答案略

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

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

// 此处为函数 twoWords 的定义

int main() {
```

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

运行示例（下划线部分为用户的输入）：

```
Please input the number of sentences: 3
Please input the sentences below:
What are you doing here
I am taking an exam
It might be an opportunity for review
The longest word is: opportunity
The shortest word is: I
```

答案略

6．编写函数 bool check(char *name)，判断字符串 name 是否为合法的变量名。

答案略

7. 用户输入一个非负整数数组元素个数 n 和每个元素，程序将这些元素按升序排列输出并用空格分隔，且奇数在前，偶数在后。

输入样例：

```
10
10 9 8 7 6 5 4 3 2 1
```

输出样例：

```
1 3 5 7 9 2 4 6 8 10
```

答案略

8. 设计函数 void printStart(char *str, char ch)，已知字符串 str 中只包含由小写字母组成的单词和空格，且首尾字符都不是空格，相邻两个单词之间有且只有一个空格，要求在屏幕输出 str 中所有以小写字母 ch 开头的单词，且以空格分隔。

答案略

9. 已知如下向量类 vect 的部分代码，私有成员包含向量长度（size）和向量数据（data），向量中保存的数据类型为 double。

```
class vect {
    double *data;
    int size;
  public:
    vect(double *, int n);
```

```
        ~vect();
    };
```
以及函数原型：
```
    double vec_mul(const vect&, const vect&);
```
该函数能够计算两个向量的内积，即两个向量对应分量的乘积的和，可以认为两个向量的长度是一致的，无需判别长度的合法性。请实现 vec_mul 函数以及 vect 类的构造函数和析构函数，并且根据需要添加其它的 public 函数，但不得增加其它任何类型的函数，使下方的测试代码可以正常运行。
```
    int main() {
        double a[5] = {0.1,0.3,0.4,1.2,0};
        double b[5] = {1.0, -2.3, 4.5, -0.7, -1.2};
        double c[2] = {2.3, 1.4};
        vect vec1 = vect(a, 5);
        vect vec2 = vect(b, 5);
        vect vec3 = vect(c, 2);
        double mul1 = vec_mul(vec1,vec2);
        double mul2 = vec_mul(vec3,vec3);
        cout << mul1 << endl << mul2 << endl;
        return 0;
    }
```

答案略

10. 输出能被 11 整除且不含有重复数字的所有 3 位数，并统计其个数

答案略

11. 根据以下 main 函数编写函数 stringCombine
```
int main()
{
    char s1[100], s2[100];
    char *s3;
    cin.getline(s1, 100);
    cin.getline(s2, 100);
    s3=stringCombine(s1, s2);
    cout<<s3<<endl;
    delete []s3;
    return 0;
}
```

输入	输出
abcde fghijklmn	afbgchdieklmn
12345 6789	162738495

答案略

12. 在已给代码段的基础上实现圆和矩形两种类
```
class shape
{
public:
```

```
    virtual double getarea()=0; //  面积
    virtual double getperim()=0; //  周长
}

//你的代码将出现在此处

int main()
{
    Circle c(10);
    Rect r(4, 5);
    shape *s[]={&c, &r};
    for(int i=0; i<2; i++)
        cout<<"面积="<<s[i]->getarea()<<"\t  周长="<<s[i]->getperim()<<endl;
        return 0;
}
```

答案略

13. 创建类 sortAray，该类对象可以存储一个整数数组，该数组中的元素自动保持升序，使得以下代码能够得到正确的输出并且不会有内存泄漏

```
int main()
{
    int n, data;
    cin>>n;
    sortArray arr(n);
    for(int i=0; i<n; i++)
    {
        cin>>data;
        arr.add(data);
    }
    cout<<arr<<endl;
    return 0;
}
```

输入	输出
5 2 4 1 6 3	1 2 3 4 6
10 0 8 6 4 2 1 3 5 7 9	0 1 2 3 4 5 6 7 8 9

答案略