

### 选择题

计算机基础：

(1) 十进制数 14 对应的二进制数( )

- A. 1111    B. 1110    C. 1100    D. 1010

(2) 二进制数  $1011+1001=( )$

- A. 10100    B. 10101    C. 11010    D. 10010

(3) 十进制数 123 变换为等值的二进制数是( )

- A. 1111000    B. 1111010    C. 1111011    D. 1111100

(4) 二进制数  $10101011B$  变换为等值的十进制数是( )

- A. 17    B. 161    C. 21    D. 171

(5)十六进制数 ABH 变换为等值的十进制数是( )

- A. 17    B. 161    C. 21    D. 171

(6) 下列个数值中，最大的数是( )

- A. 7DH    B. (174)Q    C. 123    D. 1111100B

(7) 计算机系统中采用补码运算的目的是( )

- A. 与手工运算方式保持一致    B. 减少存储空间  
C. 简化计算机的设计    D. 提高运算的精度

(8) 把一个汉字表示为两个字节的二进制码，这种编码称为( )码

- A. 五笔字型    B. 机内    C. 拼音    D. ASCII

(9) 计算机中存储数据的最小单位是( )

- A. 字节    B. 位    C. 字    D. KB

(10) 一个浮点法表示的数由( )两部分组成

- A. 指数和基数    B. 尾数和小数  
C. 阶码和尾数    D. 整数和小数

(11) 计算机内部采用的数制是( )

- A. 二进制    B. 八进制    C. 十进制    D. 十六进制

(12) 6 位二进制数能表示的最大十进制数是( )

- A. 32    B. 31    C. 64    D. 63

(13) n 位二进制数能表示的最大十进制数是( )

- A.  $2^n - 1$     B.  $2^n$     C.  $2^{n-1} - 1$     D.  $2^{n-1}$

(14) 计算机的存储器中，一个字节由( )个二进制位组成

- A. 32
- B. 31
- C. 64
- D. 63

(15) 32 位计算机系统指的是计算机的( )是 32 位

- A. 内存
- B. 硬盘
- C. 字长
- D. 位长

(16) 在计算机中采用二进制，是由于( )

- A. 硬件成本低
- B. 系统稳定
- C. 运算简单
- D. 上述 3 个原因

(17) 下面最小的一个数是( )

- A. 11011001B
- B. 75
- C. (37)o
- D. (2A)h

(18) 二进制数 1001101011 转换为等值的八进制数是( )

- A. 4651
- B. 1153
- C. 9AC
- D. 26B

(19) 二进制数 10011010111 转换为等值的十六进制数是( )

- A. 4651
- B. 1153
- C. 9AC
- D. 26B

(20) 在一个非零的二进制数后增加一个 0 后的数是原来的( )倍

- A. 1
- B. 2
- C. 3
- D. 4

(21) 下列叙述中，正确的是( )

- A. 任何一个十进制小数都可以用有限位二进制小数精确地表示出来
- B. 在一种数制中，最小的数码是 0，而最大的数码是基数本身
- C. 按字符的 ASCII 码比较，A 比 g 大
- D. 负数的补码的补码是其原码

(22) 字符的 ASCII 码的表示方法是，使用 8 位二进制码并且( ).

- A. 最低位为 0
- B. 最低位为 1
- C. 最高位为 0
- D. 最高位为 1

(23) 下列无符号数中最大的是( )

- A. 101
- B. (66)h
- C. (145)o
- D. (01100101)b

(24) 以下 4 个数未标明属于哪一种数制，但是可以断定( )不是八进制数

- A. 1234
- B. 5678
- C. 1111
- D. 1103

(25) 与十进制数 230 等值的十六进制数是( )

- A. E6
- B. F6
- C. E2
- D. F2

(26) 已知字符 0 的 ASCII 码是 48，则字符 9 的 ASCII 码是( )

- A. 32
- B. 9
- C. 39
- D. 57

(27) 十进制数 -100 的 8 位二进制补码是( )

- A. 10011100    B. 11100100    C. 00011011    D. 10011011

(28) 在表示存储器的容量时, 1MB 的准确含义是( )字节

- A. 1000K    B. 1024\*1024    C. 512\*512    D. 2048\*2048

(29) 一台微型计算机的内存容量 2G, 指的是该微机的内存是 2G 个( )

- A. 位    B. 字    C. 字节    D. 块

(30) 已知字母 a 的 ASCII 码是 97, 则字母 f 的 ASCII 码是( )

- A. 100    B. 101    C. 102    D. 103

参考答案 :

b a c d d

a c b b c

a d a d c

d c b d b

d c b b a

d a b c c

基本语句 :

(1)以下各选项中,非法的变量名是()

- A. No\_1    B. No123  
C. 2name    D. sumOfStudent

(2)下面定义变量语句中,错误的是()

- A. int x(10), y(10);    B. int x=y=10;  
C. int x=10, y=10;    D. int x=10, y==x;

(3)运算符优先级按由高到低排序正确的是 ()

- A. ==, ||, !=, %  
B. ==, %, ||, !=  
C. %, !=, ||, ==  
D. ||, !=, %, ==

(4)在 C++ 中,要求操作数必须是整形的运算符是 ()

- A. %    B. &&    C. ./    D. <=

(5)设 i、j、k 都是变量,下面不正确的赋值表达式是 ()

- A. i++  
B. i=j=k  
C. i=j==k  
D. i+j=k

(6)若 m、n 为整型,x 为实型,ch 为字符型, 下列赋值语句中正确的是 ()

- A.m+n=x;
- B.m=ch+n ;
- C.x=(m+1)++;
- D.m= x%n;

(7)若整型变量 a、b、c、d、m、n、k 的值均为 1,运行表达式(m=a>=b)&&(n=c<=d)|| (k=0) 后, m、n、k 的值是()。

- A.0、0、0
- B.1、1、1
- C.1、1、0
- D.0、1、1

(8)整型变量 m、n 的值都为 3,下列表达式中结果为 0 的是()。

- A. m&&n
- B. m&n
- C. m||n
- D. m^n

(9)下面语句的输出结果是()。

```
int x=6, y= 3;
cout<<(x++,--y,x%y, x/y)<<' , ' <<x << ' ' <<y <<endl ;
```

- A.3,7,2
- B.1,6,3
- C.0,6,3
- D.4,7,2

(10)变量 x 表示成绩,C++ 中表示  $0 \leq x \leq 100$  的表达式是()。

- A.0<=x<=100
- B.0<=x&& x<=100
- C.0<=x!<=100
- D.0<=x||x<=100

(11)下面说法中正确的是(.

- A.空语句就是一个空行
- B.空语句是什么也不输出的语句
- C.复合语句就是多条语句
- D.复合语句逻辑上是一条语句

(12)C++ 中,常量 99.78 默认的数据类型是().

- A. int
- B.float
- C. double

D. long double

(13)C++中常量 123 默认的数据类型是( )

- A. int
- B. float
- C. double
- D. long double

(14)表达式的  $70*43.6f + 34$  数据类型是( )

- A. float
- B. double
- C. long double
- D. 错误的表达式

(15)表达式的  $100 + 43.6 * 'a'$  数据类型是()

- A. float
- B. double
- C. long double
- D. 错误的表达式

(16)表达式的'0+30'的值为()。

- A. 30
- B. 95
- C. 78
- D. 错误的表达式

(17)表达式  $\sqrt{b * b - 4 * a * c} \geq 0 \text{ } \&\& \text{ } a \neq 0$  是( )表达式。

- A. 算术
- B. 关系
- C. 逻辑
- D. 函数

(18)下面程序运行的运行结果是()

```
#include<iostream>
using namespace std;
int main()
{
    char a='1';
    cout<<++a<<"";
    cout<<a+1<<endl;
    return 0;
}
```

A. 2,2    B. 2,51    C. 50,51    D. 3,4

(19)下面程序运行的运行结果是()

```
#include<iostream>
using namespace std;
int main()
{
    int b='1';
    cout<<++b<<"";
    cout<<b+1<<endl;
    return 0;
}
```

A.2,2    B.3,51    C. 50,51   D.3,4

(20)下列表达式中的值为 1 是( )。

- A.1-'0'
- B.1-'\\0'
- C.'1'-0
- D. '\\0'-'0'

参考答案：

c b c a d  
b b d a b  
d c a a b  
c c b c b

分支循环：

(1)C++规定， else 与( )相匹配。

- A. 同一列的 if
- B. 同一行的 if
- C. 之后最近的 if
- D. 之前最近的 if

(2)下面关于循环语句的叙述中， 正确的是( )。

- A. for 循环只能用于循环次数已知的情况
- B. for 循环与 while 循环一样， 都是先执行循环体后判断条件
- C. for 循环体内不能出现 while 语句
- D. 无论哪种循环,都可以从循环体内转移到循环体外

(3)以下不正确的 if 语句形式是( )。

- A. if(x>y);
- B. if(x=0)  
    x+=y;
- C. if(x!=y)cin>>x;

```
else cin>>y;  
D. if(x<y)  
{x++; y++;}
```

(4)下面的描述，不正确的是( )。

- A. 语句 `for(i=0;;i++)` 表示无限循环
- B. 语句 `for(;;)` 表示无限循环
- C. 语句 `for()` 表示无限循环
- D. `while(1)` 表示无限循环

(5)下面关于 `break` 语句的描述中，不正确的是( )。

- A. `break` 语句用于 `if` 语句中将退出该 `if` 语句
- B. `break` 语句用于循环体内将退出该循环
- C. `break` 语句用于 `switch` 语句中将退出该 `switch` 语句
- D. `break` 语句在一个循环体内可以多次出现

(6)下面各组语句中，不是“死循环”的是( )。

- A. 

```
int i=0;  
do  
{  
    ++i;  
}while(i>=0);
```
- B. 

```
int k=100;  
while(1)  
{  
    k=k%100+1;  
    if(k==20)break;  
}
```
- C. 

```
int j, sum(0);  
for(j=1;j++)  
    sum++;
```
- D. 

```
int a=3379;  
while(a++%2+3%2)  
    a++;
```

(7)下面程序段执行后的输出结果为( )。

```
for(int i=0,m=1; i<4; i++)  
    for(int j=0; j<i; j++)  
        m+=1;  
cout<<m;  
A. 7      B. 17      C. 6      D. 16
```

(8)下面各程序段不能实现求  $n!$  功能的是( )。

- A. `int i, p, n;`

```

    cin>> n;
    for(i=1,p=1; i<=n; i++)
        p*=i;
B. int i, p, n;
    cin>> n;
    for(i=1; i<=n; i++)
        {p=1; p*=i;}
C. int i=1, p=1,n;
    cin>> n;
    while(i<= n)
        {p*=i; ++i;}
D. int i=1, p=1, n;
    cin>> n;
    do
        { p*=i; ++i;} while(i<=n);

```

(9)运行以下程序时，如果由键盘输入 9 2，则输出结果是( )。

```

#include <iostream>
using namespace std;
int main(
{
    int m,n;
    cout<<"Input m, n:";
    cin>>m>> n;
    while (m!=n)
    {
        while (m>n) m-=n;
        while(n>m) n-=m;
    }
    cout<<m<<endl;
    return 0;
}

```

- A. 3      B. 2      C. 1      D. 0

(10) 下面程序段的运行结果是( )。

```

#include <iostream>
using namespace std;
int main()
(
    int n,s=0;
    for (n=0; n++<=2;)
        s+=n;
    cout<<n<<","<<s<< endl;
    return 0;
}

```

- }
- A. 4,6     B. 3,4     C. 3,3     D. 有语法错误

(11)下面程序的运行结果为( )。

```
#include <iostream>
using namespace std;
int main()
{
    for(int a=0,x=0; !x&&a<=10; a++)
    {
        a++;
    }
    cout<<a<<endl;
    return 0;
}
```

- A. 10     B. 11     C. 12     D. 0

(12)下列语句中不正确的是( )。

- A. for(int a=1; a<=10; a++);  
B. int a=1; do{ a++; }while(a<=10)  
C. int a=1; while(a<=10) {a++;}  
D. for(int a=1; a<=10; a++)a++;

(13)下面关于 for 循环的正确描述是( )。

- A. for 循环只能用于循环次数已经确定的情况  
B. for 循环是先执行循环体语句，后判断表达式  
C. 在 for 循环中，不能用 break 语句跳出循环体  
D. 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

(14)下面的循环体执行的次数与其他不同的是( )。

- A. i=0; while(++i<=100) {cout<<i<<"";}  
B. for(i=0; i<100; i+ +) {cout<<i<<"";}  
C. for(i=100; i>1; i--) {cout<<i<<"";}  
D. i=100; do (cout<<i<<""); while(-- i>0);

(15)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：

D D B C A

B A B C A  
C B D C B

数组：

(1) 在访问数组中的某个元素时，不可以使用( )指定待访问元素的下标

- A. 整形常量
- B. 整形变量
- C. 整形表达式
- D. 浮点型常量

(2) 下面关于数组的描述中，错误的是( )

- A. 数组的长度必须在定义数组时指定，且数组中的所有元素的数据类型必须相同
- B. 如果定义一位数组时提供了初始化列表，则数组的长度可以忽略
- C. 如果定义二维数组时提供了初始化列表，则数组的列下标可以忽略
- D. 如果定义二维数组时提供了初始化列表，则数组的行下标可以省略

(3) 已知 int a[]={1,2,3,4,5};则下面叙述中正确的是( )

- A. 数组 a 的长度为 5
- B. 元素 a[1]的值为 1
- C. 使用 cin>>a;可以将从键盘上输入的整数保存在数组 a 中
- D. 使用 int b[5] = a;可以定义数组 b，并用 a 中个元素的值初始化 b 中的各元素

(4) 已知 char s[]{"hello"};则下面叙述中正确的是( )

- A. 数组 s 的长度为 5
- B. 元素 s[1]的值为'h'
- C. 使用 cin>>s;可以将从键盘上输入的字符串保存在数组 s 中
- D. 使用 int t[] = s;可以定义数组 t，并用 s 中个元素的值初始化 t 中的各元素

(5) 下面各选项中的数组定义方式，错误的是( )

- A. int a[7];
- B. const int N=7;float b[N];
- C. char c[] = "abcdef";
- D. int N=7;double d[N];

(6) 数组定义为 int a[2][3]={1,2,3,4,5,6},可以使用( )访问值为 3 的数组元素

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

(7) 已知 char s[]{"abc"};则数组 s 中最后一个元素的值为( )

- A. 'c'
- B. '0'
- C. '\0'
- D. '\n'

(8) 已知 char s[]{"南开大学"};则数组 s 的长度为( )

- A. 4
- B. 5
- C. 8
- D. 9

(9) 下面定义的一维字符型数组中，存储的数据不是字符串的为( )

- A. char s[]{"abc"};
- B. char s[]{"a",'b','c','\0'};
- C. char s[]{"a",'b','c'};
- D. char s[20] = "abc";

(10) 已知 char s[]="university";则使用语句 cout<<s[3];会在屏幕上输出( )  
A. n    B. i    C. v    D. iversity

(11) 已知 char s[][10]={"Microsoft","Visual","C++"};则语句 cout<<s,  
会在屏幕上输出()  
A. i    B. c    C. Visual    D. C++

(12) 已知 char s[][10]={"Microsoft","Visual","C++"};则语句 cout<<s[1][2],  
，会在屏幕上输出()  
A. s    B. i    C. c    D. icrosoft

(13) 下面程序的输出结果为：87，则横线处应填入()

```
#include<iostream>
using namespace std;
struct Student{
    char num[8];
    char name[10];
    int score[3];
}stu[]={{"1210101","Zhangsan",{87,97,67}},{"1210102","Lisi",{92,86,79}}};
int main(){
    cout<<____;
    return 0;
}
```

A. stu[1].score[1]    B. stu[0].score[0]  
C. stu[1].score[0]    D. stu[0].score[1]

(14) 已知 enum Color{Red,Green,Blue};Color co;则下列语句正确的是( )  
A. co = 0;    B. co = Blue;  
C. co = Green + 1;    D. co++;

参考答案：

d c a c d  
b c d c c  
d a b b

函数：

(1)下列叙述中,正确的是()  
A.在一个函数的函数体中可以定义另一个函数,但不能调用另一个函数  
B.在一个函数的函数体中可以调用其他函数,但不能调用自己  
C.在一个函数的函数体中既可以定义另一个函数,也可以调用另一个函数

D.在一个函数的函数体中既可以调用其他函数,也可以调用自己

(2)函数定义中,下列有关 return 语句的描述错误的是().

- A.函数定义中可以没有 return 语句
- B.函数定义中可以有多个 return 语句,但只能执行其中一个 return 语句
- C.函数定义中,一个 return 语句可以返回多个值
- D.只要和函数类型一致,return 后面可以是常量、变量和任一表达式

(3)在 C++ 中,函数原型中可以省略().

- A.函数类型
- B.函数名
- C.形参类型
- D.形参名

(4)有关函数的形参和实参的描述,错误的是()。

- A.函数调用时传入的参数称为实参
- B.函数定义时给出的参数称为形参
- C.形参和实参可以同名
- D.在函数体中修改形参,则相应实参的值也会改变

(5)下列函数原型中,错误的是( ).

- A.int fun(int, int);
- B void fun(int x,y);
- C.int fun(int x,int y);
- D void fun();

(6)已知  $x=3, y=5$ ,则函数调用语句  $\text{fun}(2*y-1,(++x,y))$  中第二个实参的值为 ()

- A.9
- B.3
- C.4
- D.5

(7)下列有关内联函数的描述中,错误的是 ()

- A.内联函数必须在定义外加上 inline 关键字,否则就是普通的函数
- B.内联函数必须是一个小函数,不能包含循环, switch 等语句
- C.一个函数中如果包含循环, switch 等语句,则将其定义为内联函数时编译器会报错
- D.在编译程序时,系统会直接将调用内联函数的地方用内联函数中的语句做等价替换,从而省去运行程序时函数调用所额外消耗的时间。

(8)下列函数原型中,正确的是( ).

- A void fun(int a=10,int b,int c);
- B void fun(int a=10,int b=5,int c);
- C void fun(int a,int b=5,int c=8);
- D void fun(int,int=5,int);

(9)下列对有关带默认形参值的函数的描述中,正确的是()。

- A.只能在函数定义时设置默认形参值
- B.只能在函数声明时设置默认形参值
- C.函数调用时,必须使用默认的形参值,不能给出新的实参值

D.默认形参值必须严格按照从右至左的顺序进行指定

(10)默认形参值不可以是()

- A.局部变量
- B.全局变量
- C.静态全局变量
- D.函数调用

(11)已知 fun() 函数的函数原型为 void fun(int x; double y=3.5, char z='#'); 则下面的函数调用中, 不合法的调用是()。

- A.f(1);
- B.f(2,4.2);
- C.f(3,3.7,'\*')
- D.f(0, , '#')

(12)对于重载函数, 程序在调用时根据()能够区分到底要调用哪个函数.

- A.函数名
- B.函数类型
- C.参数个数或参数类型
- D.以上都可以

(13)下列函数声明中, 为重载函数的一组是().

- A. void fun(int);void fun(double=5.0);
- B. void fun(int);int fun(int=5);
- C. int f1(int);int f2(int, int);
- D. void fun(int);int fun(int, int=3);

(14)如果需要一个变量只在某个函数中可以使用, 且每次执行函数时都重新定义并初始化该变量, 那么这个变量应定义为()

- A.局部变量
- B.全局变量
- C.静态局部变量
- D.静态全局变量

(15)如果需要一个变量来记录函数的调用次数, 那么这个变量不能定文为()

- A.局部变量
- B.全局变量
- C.静态局部变量
- D.静态全局变量

(16)下列关于函数的描述中, 错误的是()。

- A.在一个源文件中定义的外部函数可以在其他源文件中调用, 静态函数不可以
- B.使用 static 关键字可以将一个函数定义为静态函数

- C.在一个源文件中定义的内联函数不能在其他源文件中调用
- D.函数的外部声明中可以省略 extern 关键字

(17)自定义头文件中一般不包含( )

- A.数据类型的定义
- B.全局变量的定义
- C.符号常量的定义
- D.内联函数的定义

(18)下列有关编译预处理命令的描述中,错误的是()

- A.编译预处理命令都是以#开头
- B.编译预处理命令在编译之前进行处理
- C.编译预处理命令#define X 5+1, 会将程序中出现的 X 替换为 6
- D.编译预处理命令可以放在程序的开头、中间或末尾

(19)下列有关文件包含的描述中,错误的是 ()

- A.#include 后面指定包含的文件可以是系统的头文件
- B.#include 后面指定包含的文件可以是自定义的头文件
- C.如果包含的头文件名用尖括号括起, 则会先在当前工作目录下搜索头文件
- D.如果包含的头文件名用双引号括起, 则会现在当前工作目录下搜索头文件

参考答案 :

d c d d b  
d c c d a  
d c a a a  
c b c c

引用和指针 :

- (1) 已知 int a=5, b, \*p = &a;, 则下列语句中正确的是 ( )。
  - A. &b = a;
  - B. b = \*a;
  - C. \* p = &b;
  - D. b=p;
- (2) 已知 int a[10], \*p = a;, 则以下列各个选项中对数组元素 a[2] 访问错误的是 ( )。
  - A. \*(a+2)
  - B.p[2]
  - C. \*(p+2)
  - D. p+2
- (3) 已知 int a [] = {1, ,2, 3, 4, 5}, \*p = a; , 则以下选项中对数组元素访问错误的是 ( )。
  - A. \*(p+2)
  - B. a[5]
  - C. a[3]
  - D. p[p-a]
- (4) 已知 int \* p[5];, 则 p 是 ( )。
  - A. 指针数组
  - B. 函数指针变量
  - C.指向行的指针变量
  - D.一级指针变量
- (5) 已知 int (\* p)( );, 则 p 是 ( )。

- A. 指针数组    B. 函数指针变量    C. 指向行的指针变量    D. 一级指针变量

(6) 已知 `int a[3][2] = {{0, 1}, {2, 3}, {4, 5}}, (* P)[2] = a+1;`, 则 `p[1][1]` 的值是 ( )。  
A. 2    B. 3    C. 4    D. 5

(7) 已知程序段

```
int a[3][4], * p[3] = { a[0], a[1], a[2]}, **pp=p, i;
for (i=0; i < 12; i++)
    a[i/4][i%4] = i;
```

则 `pp[1][2]` 的值是 ( )。

- A. 3    B. 4    C. 5    D. 6

(8) 已知 `int a[] = {1, 2, 3, 4, 5}, *p = a;`, 则以下各选项中值为数组元素地址的是 ( )。  
A. `*(a+2)`    B. `&(a+3)`    C. `p+3`    D. `*(&a[3])`

(9) 已知程序段:

```
int a[] = {1, 2, 3, 4, 5}, *p;
p = a+1;
*(p+2) += 2;
cout << *p << ',' << *(p+2) << endl;
```

则输出的结果是 ( )。

- A. 1, 5    B. 2, 6    C. 1, 6    D. 2, 5

(10) 已知 `int a[5] = {10, 20, 30, 40, 50}, * p1, * p2; p1 = &a[2]; p2 = &a[4];`, 则 `p2 - p1` 的值是 ( )。  
A. 2    B. 3    C. 20    D. 30

(11) 对于相同类型的指针变量, 不能进行的运算是 ( )。

- A. =    B. \*    C. -    D. >

(12) 已知 `int *p = new int[5];`, 若堆内存分配成功, 则指针 `p` 所指向的内存空间大小为 ( ) 字节。  
A. 5    B. 10    C. 20    D. 不确定

(13) 假设堆内存分配均成功, 则下面程序段完全正确的是 ( )。

- A. `int * p = new int(3); cout << *p; delete p;`  
B. `int *p = new int[3]; for (int i = 0; i < 3; i++) *p ++ = i; delete []p;`  
C. `int *p = new int[3]; for (int i = 0; i < 3; i++, p++) (*p = 1; cout << *p; } delete []p;`  
D. 以上程序段均正确

(14) 已知 `int a = 2, b = 3, &r = a; r = b; r = 5; cout << a << "," << b;`, 则输出结果为 ( )。  
A. 2, 5    B. 2, 3    C. 5, 3    D. 5, 5

- (15) 已知 `int a = 10, * p , *&rp = p; rp = &a; * p+ = 20; cout<<a;`， 则输出为 ( )。  
A. 10      B. 20      C. 30      D. 程序有错误
- (16) 已知 `char a[ ] = "abcd",* p= a+2;`， 则语句 `cout<<p;` 会在屏幕上输出 ( )。  
A. bcd      B. c      C. cd      D. d
- (17) 已知 `char str[20] = "mybook", *p = str+2;;` 则以下各选项中输出结果为字符 b 的是 ( )。  
A. `cout<<p;`      B. `cout<<p[0];`      C. `cout<<str;`      D. `cout<<str+2;`
- (18) 已知函数原型 `int fun( int, int );` 要定义一函数指针变量 p 使得 `p=fun;`成立， 则函数指针变量 p 的定义语句为 ( )。  
A. `int (*p)(int, int );`      B. `int *p (int, int) ;`      C. `int *p[int ,int] ;`      D. 以上选项都不对
- (19) 已知 `int a[2][3], b = fun(a);`， 则 `fun( )`函数原型为 ( )。  
A. `void fun(int (* p)[2]);`      B. `void fun (int(*p)[3]);`  
C. `int fun(int (*p)[2]);`      D. `int fun (int (*p)[3]);`
- (20) 已知 `int *a[3], b = fun(a);`， 则 `fun( )`函数原型为 ( )。  
A. `void fun(int (*p)[3]);`      B. `void fun(int **p);`      C. `int fun(int (*p)[3]);`  
D. `int fun(int *p[ ] );`
- (21) 已知函数调用 `char str[2][10]={ "abc", "def"}; fun(str);`， 则下列给出的 `fun( )`函数原型中正确的是 ( )。  
A. `void fun(char (*p)[2]);`      B. `void fun(char (*p)[10]);`  
C. `void fun(char *p[2]);`      D. `void fun(char **p);`
- (22) 已知 `int a[2][3]; fun(a);`， `fun` 函数的形参变量名为 p， 则在 `fun( )`函数体重通过 `sizeof(p)` 计算得到的结果为 ( )。  
A. 4      B. 6      C. 12      D. 24
- (23) 已知函数原型 `void fun( const int &a);`， 则下列 `fun( )`函数调用正确的是 ( )。  
A. `int x= 3; fun(x);`      B. `const int y=3; fun(y);`      C. `fun(3);`      D. 以上 3 中都正确
- (24) 已知 `int *p; fun(p);`， 其中 `fun` 没有返回值， 且其形参定义为引用调用方式，则下列给出的 `fun` 函数原型中正确的是 ( )。  
A. `void fun(int a[ ]);`      B. `void fun(int * &a);`      C. `void fun(int &a[ ]);`  
D. `void fun(int &a);`
- (25) 已知函数定义 `void fun(int &a, int b){ b++; a++;}`， 则执行 `int x=2, y= 3; fun(x, y);`后， 变量 x, y 的值分别为 ( )。  
A. 2, 3      B. 3, 4      C. 2, 4      D. 3, 3

(26) 已知 char \*s = "mybook";, 则 strlen(s)的值为 ( )。  
A. 4      B. 6      C. 7      D. 不确定

(27) 已知 char \*s = "mybook"; , 则下列语句正确的是 ( )。  
A. strcpy(s, "hello");      B. strcat(s, "hello");      C. s= "hello"      D. 以  
上均不正确

参考答案：

b d b a b  
d d c b a  
b c a c c  
c b a d d  
b a d b d  
b c