

第 5 章 指针的应用

批注 [11]: 文档中标注为红色的题较难！可自行编制程序查看运行结果，然后思考原因。

二、单项选择题

1. 下列关于指针运算的描述错误的是 (B)。
(A) 指针变量是用于存储变量地址的变量
(B) 指针变量是用于存储变量值的变量
(C) 在一定条件下，指针变量可以为空值
(D) 可以使用指针变量来访问数组
2. 若已经定义了 a 为 int 型变量，则对指针变量 p 的定义及初始化正确语句是 (D)。
(A) int *p=a; (B) int p=a; (C) int *p=*a (D) int *p=&a;
3. 若有定义：int x=0, *p=&x;，则语句：cout<<*p; 的输出结果是 (D)。
(A) 随机值 (B) p 的地址 (C) x 的地址 (D) 0
4. 设有定义语句：int n=0, *p=&n, *q; 则以下选项中，正确的语句是 (D)。
(A) p=1; (B) p=q; (C) *p=*q; (D) *p=5;
5. 已知一运行正常的程序中有下面的语句，由此可知，变量 a 和 b 的类型分别是 (D)。

int *p2=&x, *p1=a; *p2=*b;

- (A) int 和 int (B) int *和 int (C) int 和 int * (D) int *和 int *
6. 设有如下的程序段，程序段运行后输出的结果是 (B)。

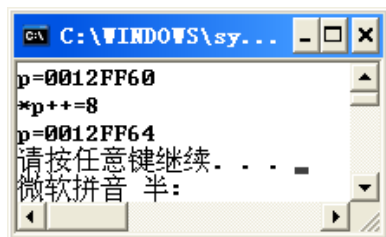
```
{ int a=1,b=3,c=5;
  int *p1=&a,*p2=&b,*p=&c;
  *p=*p1>(*p2);
  cout<<c<<endl;
}
```

7. 设有如下程序段：

```
int x=8,*p=&x;
```

```
cout<<*p++<<endl; 输出的值为 ( A )。
```

- (A) 8 (B) 9 (C) 8 的地址 (D) 9 的地址



```
#include <iostream>
using namespace std;
void main()
{
  int x=8,*p=&x;
  cout<<"p="<<p<<endl;
  cout<<"*p++="<<*p++<<endl;
  cout<<"p="<<p<<endl;
}
```

批注 [12]: 指针变量 p 有值，可以把 p 的值赋给 q，但由于 q 没有初值所有不能进行赋值、取值等操作

批注 [13]: 在定义语句中，对指针变量 *p1 进行初始化，若有复制成功，那么 a 和 p1 类型一样，所以 a 为 int *，在第 2 个语句中，是把 b 的值赋给指针变量 p2，取值运算是针对指针变量的，故 b 也是 int * 型。

批注 [14]: *P++ 是先取值再自增，即等价于：*P，P++；P 指向下一个存储空间

8. 设有如下程序段：

```
int x=8,*p=&x;
```

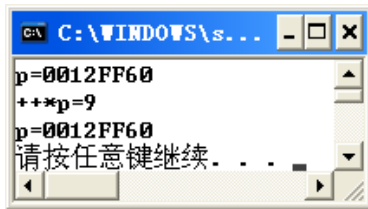
```
cout<<++*p<<endl; 输出的值为 ( B )。
```

(A) 8

(B) 9

(C) 8 的地址

(D) 9 的地址



```
#include <iostream>
using namespace std;
void main()
{
    int x=8,*p=&x;
    cout<<"p="<<p<<endl;
    cout<<"++*p="<<++*p<<endl;
    cout<<"p="<<p<<endl;
}
```

批注 [15]: 先取指针变量 p 所指向的内存单元的值, 即 8, 再对取到的结果进行加 1 的操作, 即:

++ (*p)

9. 设有定义语句: int *point, a=4; 和 point=&a; 下面均代表地址的一组选项是 (C)。

(A) a, point

(B) &a, *point

(C) point, &a

(D) a, *point

10. 设有定义语句: int a[10], *p=a; 对数组元素正确使用的语句是 (C)。

(A) a[p]

(B) p[a]

(C) *(p+2)

(D) p+2

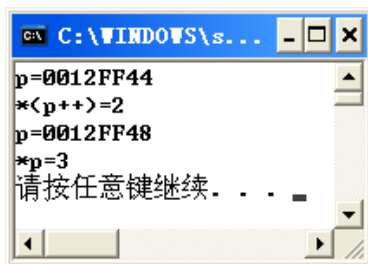
11. 设有定义语句: int a[10]={0,1,2,3,4,5,6,7,8,9}, *p=a; 则数值不为 3 的表达式是 (C)。

(A) a[3]

(B) p[3]

(C) p+=2, *(p++)

(D) p+=2, *(++p)



```
#include <iostream>
using namespace std;
void main()
{
    int a[10]={0,1,2,3,4,5,6,7,8,9}, *p=a;
    p+=2;
    cout<<"p="<<p<<endl;
    cout<<"*(p++)="<<*(p++)<<endl;
    cout<<"p="<<p<<endl;
    cout<<"*p="<<*p<<endl;
}
```

批注 [16]: 数组元素的 4 种表示方法:

a[i], p[i], *(p+i) *(a+i)

数值为 3 的是第 4 个元素, 其下表为 3.

C 答案, p+=2,是指针变量 p 从当前位置 (第 1 个元素) 往后 (往下) 移动 2 个数据的位置 (即到第 3 个元素位置), 语句 *(p++), 相当于先取出 p 指向的内存单元的内容, 然后 p 再往后移动一个位置。

D 答案, *(++p) 是先对 p 移动一个位置, 再取值, 即 p 从第 3 个元素位置移动到第 4 个元素位置, 取值即为 3

12. 设 int x[] = {1, 2, 3, 4, 5, 6}, *p=x; 则数值为 3 的表达式是 (B)。

(A) p+=2, **p

(B) p+=2, *p++

(C) p+=3, *p

(D) p+=2, ++*p

13. 设有定义语句: int a[5], *p=a; 则下列描述错误的是 (B)。

(A) 表达式 p=p+1 是合法的

(B) 表达式 a=a+1 是合法的

(C) 表达式 p-a 是合法的

(D) 表达式 a+2 是合法的

14. 有下面语句 int a[10]={10,9,8,7,6,5,4,3,2,1}, *p=a; 则数值为 2 的表达式是 (C)。

(A) a[9]

(B) *p[8]

(C) *(a+8)

(D) p+8

16. 设有如下程序段:

```
char *s="abcde";
```

```
s+=2;
```

```
cout<<*s<<endl; 输出的值为 ( B )。
```

(A) cde

(B) 字符 c

(C) 字符 c 的地址

(D) 无确定的输出结果

17. 设有如下程序段:

批注 [17]: B: p+=2, 指针变量 p 指向第 3 个元素, *p++ 是先取 p 指向的内存单元的值 (3), 在把指针变量 p 移向下一个存储位,

批注 [18]: 数组在内存中的首地址可以用数组名表示, 但它是一个常量, 故不能通过赋值运算对常量地址进行修改。也不能进行自增、自减运算。

批注 [19]: 数组元素的 4 种表示方法: a[i], p[i], *(p+i) *(a+i) 值为 2 的是第 9 个元素, 下标为 8, C 正确, p+8 标识地址, p[8] 表示第 9 个元素, *p[8] 是错误的表示方式

批注 [110]: 字符数组跟普通数组一样的理解, 只不过里面存储的是一个字符而已。

字符型的指针变量 s 指向串的第一个元素, s+=2, 则指向第 3 个元素, 最后取值 *s, 输出即为字符 c

char str[]="hello",*p=str; *(p+5)的值是 (C)。

(A) 随机值 (B) 字母 o

(C) 字符串结束标志 '\0' (D) 字母 o 在内存的地址

第 6 章 函数的应用

二、单项选择题

- 下面哪一个 C++ 函数和数学表达式 5^3 是等价的 (C)。
(A) power(3,5) (B) pow(3,5) (C) pow(5,3) (D) sqrt(3,5)
- 下面哪一条 C++ 语句显示范围 3 到 9 之间的随机数 (B)。
(A) 1+rand()%(9-3+1) (B) 3+rand()%(9-3+1)
(C) 3+rand()%(9+3-1) (D) 9+rand()%(9+1-3)
- Visual C++ 中, 函数返回值的类型是由 (D)。
(A) return 语句中的表达式类型所决定
(B) 调用该函数时的主调函数类型所决定
(C) 调用该函数时系统临时决定
(D) 在定义该函数时所指定的函数类型所决定
- 以下叙述中正确的是 (A)。
(A) 构成 C++ 程序的基本单位是函数
(B) 可以在一个函数中定义另一个函数
(C) 主函数必须放在其他函数之前
(D) 所有被调用的函数一定要在调用之前进行定义
- 以下正确的函数头定义形式是 (B)。
(A) int fun(int x; int y) (B) int fun(int x, int y)
(C) int fun(int x, y) (D) int fun(int x;)
- 在下面的函数原型中, 存在语法错误的是 (D)。
(A) void BC(int a ,int); (B) void BC(int ,int);
(C) void BC(int ,int b=5); (D) void BC(int a ; int b);
- 若有函数原型 "int sum(int x, int y);", 则使用语句 "cout<<sum(a, b);" 调用该函数时, 实参变量 a 和形参变量 x 之间的数据传递方式为 (C)。
(A) 指针传递 (B) 引用传递 (C) 值传递 (D) 内容传递
- 以下哪一项可以作为 void 函数 cacl() 的函数原形, 并且将传递给该函数两个 int 型变量的值 (A)。
(A) void cacl(int, int); (B) void cacl(int);
(C) void cacl(int &, int &); (D) int cacl(void);
- 以下说法中哪一项是错误的 (A)。
(A) 在函数头中形参的名字必须和函数调用中实参的名字一样
(B) 在函数头中列出的形参必须包含每个参数的数据类型和名字
(C) 在函数头中, 函数名后面的一对空括号代表函数不接受任何信息
(D) 函数的语句被括在一对花括号中
- 以下哪一条 C++ 语句显示名为 address 字符串变量所包含的字符的个数 (A)。
(A) cout<<address.size(); (B) cout<<size.address();
(C) cout<<size(address); (D) cout<<address(size);

批注 [111]: 求幂次的系统函数为 pow, 不是 power!!!

批注 [112]: 定义时的形参表中, 形参的类型不能少, 形参的名字无关紧要, 甚至可以省掉!!!

批注 [113]: 定义时的形参表中, 形参的类型不能少, 形参的名字无关紧要, 甚至可以省掉!!!

11. 要求函数的功能是交换 x 和 y 中的值, 且通过正确调用返回交换结果。能实现正确交换功能的函数是 (A)。

(A) void funa (int &x,int &y)

```
{ int p;  
  p=x;x=y;y=p;}
```

(C) void func (int *x,int *y)

```
{ *x=*y;*y=*x; }
```

(B) void funb (int x,int y)

```
{ int t;  
  t=x;x=y;y=t;}
```

(D) void fund (int x,int y)

```
{ x=x+y; y=x-y; x=x-y; }
```

12. 关于引用, 下列说法中正确的是 (A)。

(A) 引用是一种特殊类型的变量, 是另一个变量的别名

(B) 引用在函数调用中传递的是变量的值

(C) 在声明引用时, 要给它另开辟内存单元

(D) 在声明引用时, 可以不进行初始化

13. 下列叙述中正确的是 (B)。

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

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

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

(D) C++程序中, main()函数必须在其它函数之后, 函数内可以嵌套定义函数

14. 下面有关函数原型(函数的声明)的说法正确的是 (D)。

(A) 函数原型与函数的定义语法是一样的

(B) 如果程序中有自定义函数, 必须有和其对应的函数原型

(C) 函数原型的参数和函数定义的参数列表在参数的个数、名称、类型上必须一一对应

(D) 函数原型和函数定义在函数名和返回类型上必须一致

15. 在 C++ 函数头定义中, 若省略了返回值的数据类型, 其默认的返回值的数据类型为 (A)。

(A) int

(B) float

(C) double

(D) void

16. 下列程序运行后的结果是 (B)。

```
#include<iostream>  
using namespace std;  
void fun(int *a, int b)  
{ int *k;  
  k=a;  
  *a=b;  
  b=*k;  
}  
void main( )  
{ int a=3,*x=&a;  
  int y=5;  
  fun(x,y);  
  cout<<a<<" "<<y<<endl;  
}
```

(A) 3 5

(B) 5 5

(C) 3 3

(D) 5 3

17. 下列程序运行后的结果是 (C)。

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

批注 [114]: 子函数的返回类型均为 void 型, 而要求函数能实现交换并返回交换的结果!
如果需要通过改变形参值从而达到改变实参值的目的, 则可以将形参设置为引用类型!!!

批注 [115]: 考试时可以自己启动 VS, 新建一个项目, 创建源文件后, 把这段代码复制粘贴进去, 编译运行看结果。

批注 [116]: 子函数实现交换的功能, 由于第 1 个参数是指针变量, 所以形成的改变会影响实参!!!, 而第 2 个参数为普通变量, 是单向传值, 即形参的改变不会影响实参的值!!! 故在主函数中调用子函数后, 实现了 3 和 5 的交换, a 随着形参的改变变成了 5, 而 y 的值保持不变

批注 [117]: 子函数第 1 个参数是字符指针变量, 即形参值的改变会影响实参, 第 2 个参数是普通变量, 单值传递, 形参的改变不会影响实参!
子函数功能: 对字符 c+1, 即得下一个字符。
主函数调用时, 即将第 1 个实参的值加 1, 即 a 的下一个字母, 为 b, 第 2 个实参为字符 A, 它的 ASCII 码值为 65, 虽然形参值加 1, 但不会影响实参的值, 所以输出为 A, b

```

void fun(char *c,int d)
{
    *c=*c+1;d=d+1;
}
void main( )
{
    char a='A',b='a';
    fun(&b,a);
    cout<<a<<"    "<<b<<endl;
}

```

(A) B a, (B) a B (C) A b (D) b B

18. 下列程序运行后的结果是 (A)。

```

#include<iostream>
using namespace std;
void fun(char *a, char *b)
{
    a=b;
    (*a)++;
}
void main( )
{
    char c1='A', c2='a', *p1, *p2;
    p1=&c1;
    p2=&c2;
    fun(p1,p2);
    cout<<c1<<c2<<endl;
}

```

(A) Ab (B) aa (C) Aa (D) Bb

19. 以下不正确的说法是 (C)。

- (A) 在 C++ 中，一个程序中的函数名不一定是唯一的
- (B) 在 C++ 中，定义函数时，在返回值数据类型的前面加上关键字 `inline`，这种函数称为内联函数
- (C) 在 C++ 中，不允许有两个及以上的函数使用同一个函数名
- (D) 在定义函数时，还可以对形式参数进行初始化，即给该形式参数指定默认值。

20. 以下正确的说法是 (B)。

- (A) 全局变量如果没有指定初值，则其初值为 0
- (B) 静态变量如果没有指定初值，则其初值为 0
- (C) 局部变量如果没有指定初值，则其初值为 0
- (D) 函数中的静态变量在函数每次调用时，都会重新设置初值

21. 下面哪一个 C++ 函数和数学表达式 $|x|$ 是等价的 (B)。

(A) `pow(x)` (B) `abs(x)` (C) `sqrt(x)` (D) `power(x)`

22. 下面哪一条 C++ 语句显示范围 5 到 50 之间的随机数 (D)。

(A) `1+rand()%(50+5-1)` (B) `5+rand()%(50+5-1)`
 (C) `1+rand()%(50-5+1)` (D) `5+rand()%(50-5+1)`

23. Visual C++ 中，函数返回值的类型是由 (A)。

- (A) 定义该函数时所指定的函数类型决定
- (B) `return` 语句中表达式的类型决定
- (C) 主函数的类型决定
- (D) 调用该函数的主调函数类型决定

批注 [118]: 子函数功能：把形参 b 的值赋给 a (因为 a, b 均为字符指针变量，所以 a, b 的值为地址值)，即指针变量 a, b 指向同一个内存空间。之后取出指针变量 a 指向的内存单元的内容，再对结果加 1

批注 [119]: 主函数中，指针变量 p1 和 p2 指向 c1 和 c2 在内存中存储单元，调用子函数，相当于 p1 和 p2 均指向字符 'a' 所在的存储空间，取值后得到 a，对 a 加 1 得到字符 b

24. 以下叙述中正确的是 (A)
- (A) 构成 C++ 程序的基本单位是函数
 - (B) 函数允许嵌套定义
 - (C) 在 C++ 程序中, 主函数必须位于其他函数之前
 - (D) 函数在被调用之前必须进行函数原型的声明
25. 以下正确的函数头定义形式是 (B)。
- (A) void f(int a; int b)
 - (B) double f(double a, double b)
 - (C) int f(int a, b)
 - (D) char f(char a;)
26. 下面的函数原型中, 存在语法错误的是 (D)。
- (A) int pro(int a, int);
 - (B) int pro(int, int);
 - (C) int pro(int , int b=5);
 - (D) int pro(int a; int b);
27. 若有数据原型 “double sot(double m, double &n);”, 则使用语句 “cout<<sot(x,y)” 调用该函数时, 实参变量 y 与形参变量 n 之间的数据传递方式为 (B)。
- (A) 指针传递
 - (B) 引用传递
 - (C) 值传递
 - (D) 内容传递
28. 以下函数原型中, (C) 表示调用该函数时将传递给该函数两个 int 型变量的值。
- (A) void xy(int a, int &b);
 - (B) void xy(int &a, int b);
 - (C) void xy(int, int);
 - (D) void xy(a, b);
29. 下列说法错误的是 (A)。
- (A) 每个实参的名字应该和形参的名字相同
 - (B) 每个实参的数据类型应该和对应形参的数据类型相匹配
 - (C) 实参的数量应该和形参的数量一致
 - (D) 当按值传递给函数时, 该函数将接收的每项值存储在单独的内存单元中
30. 以下 (D) 语句的功能是输出名为 str1 字符串变量所包含的字符个数。
- (A) cout<<size.str1();
 - (B) cout<<size(str1);
 - (C) cout<<str1(size);
 - (D) cout<<str1.size();
31. 要求函数的功能是交换变量 a、b 中的值, 且通过正确调用返回交换结果。能实现正确交换功能的函数是 (D)。
- (A) void f(double a, double b)
{ int x;
x=a; a=b; b=x;}
 - (B) void f(double &a, double &b)
{ int y;
y=a; a=y; b=a;}
 - (C) void f(double a, double b)
{ int m;
m=b; b=a; a=m;}
 - (D) void f(double &a, double &b)
{ int z;
z=b; b=a; a=z;}
32. 下列说法错误的是 (C)。
- (A) 引用在函数调用中传递的是变量的地址
 - (B) 在声明引用时, 不需要给它另开辟内存单元
 - (C) 声明引用时, 可以不进行初始化
 - (D) 引用是一个变量的别名
33. 下列描述正确的是 (D)。
- (A) 程序中必须有 main() 函数, 可以根据需要写几个 main() 函数。
 - (B) 程序中必须有 main() 函数, 而且规定 main() 函数只能写在所有的函数最前面。

- (C) 程序中必须有 `main()` 函数，可以根据需要在 `main()` 函数内部定义其他的函数。
- (D) 程序中必须有 `main()` 函数，也可以定义其他函数，而且 `main()` 函数的位置没有限制。
34. 有关函数原型的说法正确的是（ B ）。
- (A) 函数原型在 C++ 程序中必须声明。
- (B) 函数原型和函数定义在函数名称和返回值类型上必须一致。
- (C) 函数原型的参数和函数定义的参数列表在参数个数、名称、类型上必须一一对应。
- (D) 函数原型中的每一个参数必须给出参数的类型和名称。
35. 在 C++ 函数头的定义中，如果省略了函数返回值的数据类型，则其默认的函数返回值数据类型为（ B ）。
- (A) `double` (B) `int` (C) `char` (D) `float`