

## 指针

9/15 得分 (60%)

重新测试

下一个

通过测验需要有：80% 或更高。



1 / 1 分

1。

现在要字符型的定义指针变量p，以下定义及初始值设置合法的包括哪些？

☐ `char a[] = "abc";`

`char p[] = a;`

☐ `char a[] = "abc";`

`char p[] = *a;`

☒ `char a[] = "abc";`

`char *p = a;`



正确

☐ `char a[] = "abc";`

`char p[] = &a;`

☐ `char a[] = "abc";`

`char *p = * a;`



1 / 1 分

2。

已知`int a = 3;` 现在想输出a的值，即得到输出为 3 。在不考虑代码优美性、易读性的情况下，以下操作可行的有哪些？



☒ `cout << *a;`



未选择的是正确的

☐ `cout << *&a;`



正确

&a为a的地址，\*(&a)为a的地址的内容，即a。

☐ `cout << a;`



正确

这是最正常的代码，通常情况下应该这么写。

☐ `cout << &a;`



未选择的是正确的

☐ `cout << &*a;`



未选择的是正确的



0 / 1 分

3。

已知字符串 `char a[] = "hollo,world";` 由于存在拼写错误，现在我们想让这个字符串变成"hello,world"，以下操作哪些是正确的？

☐ `a++;`

`* a = 'e';`



未选择的是正确的

☐ `*(a + 1) = 'e';`



正确



☒ `a[1] = 'e';`



正确

☐ `&(a+1) = 'e';`



未选择的是正确的

☐ `char *p = a;`

`p++;`

`*p = 'e';`



这应该被选择



0 / 1 分

4。

有`double num = 3.14; double * pi = & num;` 现在`pi`指向的地址的内容为3.14。然而我们又想要提高精度，将它变成3.14159。在不考虑代码优美性、易读性的情况下，以下操作正确的有哪些？

☐ `pi = 3.14159`



未选择的是正确的

☐ `pi[0] = 3.14159`



这应该被选择

☐ `*pi = 3.14159`



正确

正确

☐ `π = 3.14159`



未选择的是正确的



1 / 1 分

5。

已知定义了数组int a[10]; 并且a指向的地址为0x22ff44。假设整形占4个字节，那么下列哪个说法是正确的？

- ☐ a+1无合法语义
- ☐ a+1所指向的地址为0x22ff45，因此a+1不是数组元素a[1]的地址
- ☐ a+1所指向的地址为0x22ff45，因此a+1是数组元素a[1]的地址
- ☒ a+1所指向的地址为0x22ff48，因此a+1是数组元素a[1]的地址



正确

- ☐ a+1所指向的地址为0x22ff48，因此a+1不是数组元素a[1]的地址



0 / 1 分

6。

已知字符串char str[] = "hello,world"; 现在我们想输出字符串的后半部分，即",world"，以下操作正确的包括哪些？

- ☒ cout << &(str + 5) << endl;



未选择的是正确的

- ☐ cout << str[5:10] << endl;



这个选项的答案不正确

C++中没有这种写法

- ☒ for(int i = 5; i <= 10; i++)

cout << \*(str + i);

cout << endl;



正确

正确，逐个输出每个字符

☐ cout << str + 5 << endl;



这应该被选择

☐ for(int i = 5; i <= 10; i++)

cout << str[i];

cout << endl;



正确

正确，逐个输出每个字符

☐ cout << \*(str + 5) << endl;



未选择的是正确的

☐ cout << str[5] << endl;



未选择的是正确的



0 / 1 分

以下函数的输出结果是：

```
int fun( ) {
```

```
    char a[10] = {'1', '2', '3', '4', '5', '6', '7', '8', '9', 0}, *p;
```

```
    int i = 8;
```

```
    p = a + i;
```

```
    cout << p - 3 << endl;
```

```
    return 0;
```

```
7. }
```

5

不正确回答



1 / 1 分

8。  
以下函数的运行结果是：

```
int fun( ){  
  
    int a[ ]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, };  
  
    int *p = a + 5, *q = NULL;  
  
    *q = *(p + 5);  
  
    cout << *p << " " << *q << endl;  
  
    return 0;  
  
}
```



运行后报错

正确

\*q初始化为NULL，指向非法地址。语句\*q=\*(p+5)的含义相当于：请把非法地址里的内容设成\*(p+5)，故运行报错。



6 6



6 12



5 5



1 / 1 分

9。

以下函数返回a所指数组中最小的值所在的下标，其中n为数组a的大小。那么划线处应该填入：

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

- ☐ i = p
- ☐ p = & a
- ☐ p = & a[i]
- ☒ p = i

正确  
p为下标

- ☐ p = a[i]
- ☐ p = \* a[i]
- ☐ p = \* a
- ☐ p = a



1 / 1 分

10。

以下函数的返回值是什么？

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

☒ p指向的地址值



正确

☐ p自身的地址值

☐ 无意义的值

☐ p[0]这个字符



1 / 1 分

下列程序的输出结果是：

```
int b = 2;  
  
int func(int *a){  
  
    b += *a;  
  
    return b;  
  
}  
  
int main( ){  
  
    int a=2, res=2;  
  
    res += func(&a);  
  
    cout << res << endl;  
  
    return 0;
```

11. }

6



正确答案

func返回4，故res值为6

---



1 / 1 分

有如下程序段

```
int *p, a = 10, b = 1;
```

```
p = &a;
```

```
a = *p + b;
```

12。执行该程序段后, a 的值为:

11

正确答案

p为指向a的指针, a=\*p+b相当于a=a+b, 故结果为11。

---



1 / 1 分

13。

对于基类型相同的两个指针变量之间，以下哪一项操作缺乏有价值的语义？

☐ <

☐ =

☒ +



正确

☐ -

---



0 / 1 分

14。

下面程序把数组元素中的最大值放入a[0]中, 则在if 语句中的条件表达式应该是:

```
int fun( ){  
  
    int a[10] = {6, 7, 2, 9, 1, 10, 5, 8, 4, 3}, *p = a, i;  
  
    for( i = 0; i < 10; i++, p++)  
  
        if(_____)  
  
            *a=*p;  
  
    cout << *a << endl;  
  
}
```

☐ p>a



未选择的是正确的

☐ \*p>a[0]



这应该被选择

☐ p[i] > a[0]



这个选项的答案不正确

☐ \*p>\*a[0]



未选择的是正确的

☐ a[i] > p[0]



未选择的是正确的

☐ \*p[0]> \*a[0]



未选择的是正确的

☐ \*p > \*a



正确



`a[i] > a[0]`



正确



0 / 1 分

15。

以下程序片段都能为a的元素加1，并输出。考虑程序易读性、可移植性，不考虑代码的文本长度、代码行数和执行效率，你认为代码风格最良好的是哪个？



```
int a[] = {10, 21, 32, 32, 46};
```

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

```
    a[i]++;
```

```
    cout << a[i] << endl;
```

```
}
```



```
int a[] = {10, 21, 32, 32, 46};
```

```
for (int i = 0; i < 5; cout << ++a[i] << endl, i++);
```



```
int a[] = {10, 21, 32, 32, 46};
```

```
for (int * p = a; p < a + 5;)
```

```
    cout << ++ (p++) << endl;
```



```
int a[] = {10, 21, 32, 32, 46};
```

```
for (int * p = a; p < a + 5;){
```

```
    (*p)++;
```

```
    cout << *p++ << endl;
```

```
}
```



这个选项的答案不正确

不良的代码风格，`(*p)++`、`*p++`这样的语句让人难以理解，特别是后者。