指针

15/15 questions correct

Excellent!

Retake

Next (/learn/c-chengxu-sheji/lecture/yELw9/zi-fu-chuan-yu-zhi-zhen)

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;
- char a[] = "abc";
  char *p = a;

Help Center

Well done!

2.

已知int a = 3; 现在我们想输出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;

Well done!
*a无意义

- [ ] cout << a;

Well done!
这是最正常的代码，通常情况下应该这么写。

- [ ] cout << *a;

Well done!
a不是指针，不能这么写

- [ ] cout << *&a;

Well done!
&a为a的地址，*(&a)为a的地址的内容，即a。

- [ ] cout << &a;

Well done!
输出的为a的地址

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[1] = 'e';

Well done!

☐ char *p = a;

p++;

*p = 'e';

Well done!
p是指针变量，可以自加

☐ * (a + 1) = 'e';

Well done!

☐ &(a+1) = 'e';

Well done!
错误，应该将&改为*

☐ a++;

* a = 'e';

Well done!
无法对a进行自加

✔ 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?

☐ π = 3.14159

Well done!
错误，&为取地址符

☐ *pi = 3.14159

Well done!
正确

☐ pi[0] = 3.14159

Well done!
正确，虽然pi看上去不是指向数组，但确实可以通过pi[0]引用到我们需要修改的数。但是，这不是一种良好的代码风格，因为一旦使用pi[1]就会出现不可预料的后果。

☐ pi = 3.14159

Well done!
错误，pi为指针

✔ 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不是数组元素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

Well done!

 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;

Well done!
错误，通过str[5]我们得到一个字符，即','

☐ cout << str[5:10] << endl;

Well done!
C++中没有这种写法

☐ for(int i = 5; i <= 10; i++)

    cout << str[i];

cout << endl;

Well done!
正确，逐个输出每个字符

☐ cout << str + 5 << endl;

Well done!
正确，str+5指向','，并且是字符指针，对它使用cout，输出从str+5开始的字符串

☐ for(int i = 5; i <= 10; i++)

    cout << *(str + i);

cout << endl;

Well done!
正确，逐个输出每个字符

☐ cout << &(str + 5) << endl;

Well done!
错误，str+5不是一个有名变量，无法对其使用取地址操作

☐ cout << *(str + 5) << endl;

Well done!
错误，str+5指向','，故*(str+5)给出一个字符，即','

✔ 7.

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;
}
```

6789

Well done!
p - 3指向a+5，即内容为'6'的那个地址。由于p-3为字符指针，所以使用cout时输出字符串，以0（即'\0'）结尾。

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

Well done!
*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

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;
}
```

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

Well done!
p为下标

- p = * a[i]
- p = * a
- p = a[i]
- p = & a
- i = p
- p = a

10.

以下函数的返回值是什么?

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

- 无意义的值
- p指向的地址值

Well done!

- ○ p[0]这个字符
- ○ p自身的地址值

✔ 11.

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;
}
```

6

Well done!
func返回4，故res值为6

✔ 12.

有如下程序段

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

执行该程序段后, a 的值为:

11

Well done!
p为指向a的指针，a=*p+b相当于a=a+b，故结果为11。

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?

- =
- +

Well done!

- -
- <

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

Well done!

- [ ] *p[0]> *a[0]

Well done!

- [ ] a[i] > a[0]

Well done!

- [ ] *p>a[0]

Well done!

- [ ] *p > *a

Well done!

- [ ] p[i] > a[0]

Well done!

- [ ] a[i] > p[0]

Well done!

☐ *p>*a[0]

Well done!

✔ 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;
}
```

Well done!
良好的代码风格

○
```
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;
}
```

(https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc)