

## 第4章 练习

### 4.1节

4.1

```
1 105
```

4.2

```
1 1. *(vec.begin())
2 2. (*v(vec.begin()))+1
```

.运算符优先级要高与\*运算符

4.3

略

### 4.2节

4.4

```
1 16+75+0 = 91
```

4.5

```
1 1. -90+4=-86      2. -30+12=-18
2 3. 10*21%5=0      3. -10*21%4 =-2
```

4.6

```
1 if(ia % 2 == 0)
2     cout<<"偶";
3 else
4     cout<<"奇";
```

4.7

```
1 溢出：超过该类型所能表示的值
2 int a = 2^32;
```

## 4.3 节

4.8

```
1 优先级: '||' = '&&' > '=='
```

4.9

```
1  cp不能指向空地址 且 *cp的不为空字符  ==> 条件为真
```

4.10

```
1  while(cin>>ia && ia != 42){  
2  
3  }
```

4.11

```
1  if(a > b && b > c && c > d)  
2  {  
3  
4  }
```

4.12

```
1  i!=j<k  ---> i!=(j<k)
```

## 4.4 节

4.13

```
1  1. i =3; d = 3;      2. d=3.5; i =3;
```

4.14

```
1  编译不通过;  
2  i的值变为42, 且始终为真
```

4.15

```
1  int类型数据不能用int*类型赋值
```

4.16

```
1 1. if((p=getPtr()) !=0)
2     题目中那样写，相当于判断赋值操作是否成功
3
4 2. if ( i == 10)
```

## 4.5节

4.17

略（书中有具体介绍）

4.18

1 第一个值不会输出，每次输出pbeg指向的位置的后一个值

4.19

```
1 1. 指针为空 并且 指针指向的值不等于0 为真
2 2. ival不等于0 并且 ival+1不等于0 为真
3 3. 同一表达式中，运算顺序不确定
```

## 4.6节

4.20

```
1 1.合法；输出*iter，然后iter+1；      2. 合法；输出*iter + 1      3. 不合法；*(iter.empty())
2 4. 合法；(*iter).empty();      5. 不合法；++ (*iter)      6. 合法；
   iter->empty, iter++;
```

## 4.7节

4.21

```
1 void Ex4_21() {
2     vector<int> ivec = {1,2,3,4,5,6};
3     for (auto &i : ivec)
4         i = i % 2? 2*i : i;
5     cout<<ivec[0];
6 }
```

4.22

略

4.23

```
1 string p1 = s + (s[s.size() - 1] == 's' ? "" : "s")
```

4.24

略

## 4.8 节

4.25

```
1 int占32位, 说明是一个32位机器
2 先取反
3 00000000 00000000 00000000 01110001
4 11111111 11111111 11111111 10001110
5 再左移
6 11111111 11111111 11100011 1
```

4.26

```
1 unsigned int 在某些机器上不一定有32位
```

4.27

```
1 // 3 - 0x011; 7 - 0x0111
2 1. 0x011 - 3;      2. 0x0111 - 7;
3 3. 布尔表达式 true      4 true
```

## 4.9 节

4.28

```
1 利用sizeof(int)...就行, 代码略
```

4.29

```
1 1. 10*4字节 / 4字节= 10;      数组大小/数组元素大小
2 2. 8字节/4字节 = 2;      指针大小/指针所指元素大小
```

4.30

```
1 1. (sizeof x) +y;           2. sizeof(p->mem[i])
2 3. sizeof(a) < b;         4. sizeof(f());
```

## 4.10 节

4.31

1 直接替换即可

4.32

```
1 ptr初始化指向数组首地址, ix保存ptr指向数组的索引;
2 只要ix不超过size,并且ptr没有指向数组尾后元素,就执行函数体;
3 函数体执行完毕后, ix+1, ptr指向下一个数组元素
```

4.33

```
1 (someValue ? ++x, ++y : --x) , --y
```

优先级比较

## 4.11 节

4.34

```
1 1. float -> bool;          2. int->float; float -> double
2 3. char->int; int -> double;
```

4.35

```
1 1.char -> int; int -> char;    2. int -> float; 如果float类型占用的空间
   比int多, 则ui转换成float,否则ival * 1.0的结果转换成unsigned int。表达式结
   果转换成float。
2 3. 如果float类型占用的空间比int多, 则ui转换成float,否则fval转换成unsigned
   int。表达式结果转换成double。
3 4. int -> float; float -> double; double -> char
```

?

## 4.11 节

4.36

```
1 i *= static_cast<int>(d);
```

4.37

```
1 1. pv=static_cast<void *>(const_cast<string *> (ps));  
2 2. i = static_cast<int>(*pc);  
3 3. pv = static_cast<void*>(&d);  
4 4. pc=static_cast<char *>(pv);
```

尽量不使用reinterpret\_cast

## 4.12 节

---