

# 计算机科学基础 I——Homework 9

## 作业说明

- 代码阅读题请勿借助计算机直接运行程序，应通过仔细阅读与逻辑分析得出结果；
- 输出结果需严格按照程序逻辑书写：若需要换行，请将输出写成多行；若代码存在语法错误，请写 `error`；若程序无任何输出，请写 `/`；若程序陷入死循环，请写 `死循环`；
- 程序编写与调试请统一使用 Visual C++；
- 提交作业时，请将工程文件与包含代码阅读题答案的 PDF 文件一并打包，发送至邮箱：`seuiccpp@163.com`，压缩包命名格式为：学号-姓名-hw9；
- 截止时间：2025 年 12 月 28 日

## 一、判断题

1. C++ 采用编译方式处理源代码，生成的可执行程序在运行时不再依赖源代码。（ ）
2. C++ 关键字不能用作标识符。（ ）
3. 无论是 C++ 关键字还是标识符，均区分大小写。（ ）
4. 单精度浮点数 0.1 与双精度浮点数 0.1 的数值相等，仅存储精度和占用空间不同。（ ）
5. 控制浮点数输出精度时，其实际存储精度也会随之改变。（ ）
6. 将整型变量强制转换为 `double` 类型时，其存储格式会相应转换为双精度浮点格式。（ ）
7. 若函数的形参类型为 `const int`，则不能传递 `int` 类型变量作为实参。（ ）
8. 函数定义时可以没有参数列表，也可以没有返回值。（ ）

## 二、选择题

1. 若定义 `float x = 1; int y = 2;`, 则表达式 `!y && y^3 ? y-- : y++` 的值为 ( )。
  - A. 2
  - B. 0
  - C. 1
  - D. 3
2. 下列关于函数的说法错误的是 ( )。
  - A. 重载函数可以在同一个作用域内共存。
  - B. 一旦函数某个参数被赋予了缺省值，其后所有参数都必须具有缺省值。
  - C. 函数缺省参数必须在函数声明和定义中提供。
  - D. `main` 函数定义的位置没有限制，但不可以嵌套定义在其它函数内。
3. 以下关于编译与运行的说法，正确的是 ( )。
  - A. 程序只有在运行时才会进行语法检查。
  - B. 语法错误一定能通过编译。
  - C. 编译成功的程序一定可以正确运行。
  - D. 编译阶段主要检查语法和类型错误。
4. 以下说法正确的是 ( )。
  - A. 32 位编译系统中的“位”指的是 bit。
  - B. 内存空间只有在分配的时候存在。
  - C. 计算机的最小存储单位为字节。
  - D. 编译系统通过变量名访问变量。
5. 以下说法正确的是 ( )。
  - A. 变量名在程序运行期间始终占用内存空间。
  - B. 指针变量中存放的是变量的值。
  - C. 编译系统在编译阶段为局部变量分配存储空间。
  - D. 内存中每一个存储单元都有唯一的地址。

6. 关于变量和地址的描述，正确的是（ ）。
- A. 同一个变量在程序运行过程中地址可能发生变化。
  - B. 不同变量在内存中可以共用同一个地址。
  - C. 变量的地址在程序运行期间保持不变。
  - D. 地址是在程序运行结束后由系统确定的。

### 三、填空题

1.  $-3$  的 8 位二进制补码用十六进制表示为: \_\_\_\_\_
2. 二进制数 100 用八进制表示为: \_\_\_\_\_
3. 换行符对应的字符常量为: \_\_\_\_\_
4. 具有三个操作数的运算符是: \_\_\_\_\_
5. C++ 中优先级最低的运算符是: \_\_\_\_\_
6. 表示  $a$  的平方根不等于 4 的表达式为: \_\_\_\_\_
7. 已知代码: `char c = 'X'; sizeof(c - 'A')` 的值为 \_\_\_\_\_ (提示: 该知识点上课没有提到, 字符字面量的类型不是 `char`, 而是 `int`, 算术运算会发生整型提升)

### 四、程序阅读题

1. 阅读下列程序, 写出输出结果:

```
1 int a = 5;
2 int* p = &a;
3 cout << a;
```

2. 阅读下列程序, 写出输出结果:

```
1 int a = 5;
2 int* p = &a;
3 cout << *p;
```

3. 阅读下列程序, 写出输出结果:

```
1 int a = 3;
2 int* p = &a;
3 *p = 10;
4 cout << a;
```

4. 阅读下列程序，写出输出结果：

```
1 int x = 1, y = 2;
2 int* p = &x;
3 cout << *p << " ";
4 p = &y;
5 cout << *p;
```

5. 阅读下列程序，写出输出结果：

```
1 int a = 4;
2 int* p = &a;
3 int* q = p;
4 *q = 7;
5 cout << a;
```

6. 阅读下列程序，写出输出结果：

```
1 int arr[3] = {1, 2, 3};
2 int* p = arr;
3 cout << *p;
4 return 0;
```

7. 阅读下列程序，写出输出结果：

```
1 int arr[4] = {2, 4, 6, 8};
2 int* p = arr;
3 cout << p[1] << " " << arr[2];
```

8. 阅读下列程序，写出输出结果：

```
1 int arr[3] = {5, 6, 7};
2 int* p = arr;
3 p[0] = 10;
4 cout << arr[0];
```

9. 阅读下列程序，写出输出结果：

```
1 int arr[3] = {1, 2, 3};  
2 int* p = arr;  
3 int* q = &arr[1];  
4 *p = *q;  
5 cout << arr[0] << " " << arr[1];
```

10. 阅读下列程序，写出输出结果：

```
1 void change(int* p) {  
2     *p = 20;  
3 }  
4  
5 int main() {  
6     int a = 5;  
7     change(&a);  
8     cout << a;  
9     return 0;  
10 }
```

11. 阅读下列程序，写出输出结果：

```
1 void modify(int arr[]) {  
2     arr[1] = 100;  
3 }  
4  
5 int main() {  
6     int a[3] = {1, 2, 3};  
7     modify(a);  
8     cout << a[1];  
9     return 0;  
10 }
```

12. 阅读下列程序，写出输出结果：

```
1 void f(int* p) {  
2     *p = *p + 5;  
3 }  
4  
5 int main() {  
6     int x = 1;
```

```
7     f(&x);  
8     cout << x;  
9     return 0;  
10 }
```

13. 阅读下列程序，写出输出结果：

```
1 void add(int* p, int v) {  
2     *p += v;  
3 }  
4  
5 int main() {  
6     int a = 10;  
7     add(&a, 3);  
8     add(&a, -4);  
9     cout << a;  
10    return 0;  
11 }
```

14. 阅读下列程序，写出输出结果：

```
1 void g(int* p, int* q) {  
2     *p = *p + *q;  
3     *q = *p - *q;  
4 }  
5  
6 int main() {  
7     int a = 3, b = 4;  
8     g(&a, &b);  
9     cout << a << " " << b;  
10    return 0;  
11 }
```

15. 阅读下列程序，写出输出结果：

```
1 void h(int* p, int* q) {  
2     int t = *p;  
3     *p = *q;  
4     *q = t + *q;  
5 }  
6
```

```
7 int main() {
8     int x = 2, y = 5;
9     h(&x, &y);
10    cout << x << " " << y;
11    return 0;
12 }
```

16. 阅读下列程序，写出输出结果：

```
1 void k(int* p, int* q) {
2     *p = *p + 1;
3     *q = *q + 2;
4 }
5
6 int main() {
7     int a = 10;
8     k(&a, &a);
9     cout << a;
10    return 0;
11 }
```

17. 阅读下列程序，写出输出结果：

```
1 void change(int a[]) {
2     a[0] = a[0] + a[1];
3     a[1] = a[0] - a[1];
4 }
5
6 int main() {
7     int a[3] = {3, 7, 9};
8     change(a);
9     cout << a[0] << " " << a[1] << " " << a[2];
10    return 0;
11 }
```

18. 阅读下列程序，写出输出结果：

```
1 void p1(int* x) { *x += 2; }
2 void p2(int* x) { p1(x); *x *= 3; }
3
4 int main() {
```

```
5     int a = 1;
6     p2(&a);
7     cout << a;
8     return 0;
9 }
```

19. 阅读下列程序，写出输出结果：

```
1 void m(int a[], int* x, int* y) {
2     a[0] += *x;
3     a[1] += *y;
4     *x = a[0] + a[1];
5 }
6
7 int main() {
8     int a[2] = {1, 2};
9     int t = 5, s = 6;
10    m(a, &t, &s);
11    cout << a[0] << " " << a[1] << " " << t << " " << s;
12    return 0;
13 }
```

20. 阅读下列程序，写出输出结果：

```
1 void make(int x, int y, int* sum, int* diff) {
2     *sum = x + y;
3     *diff = x - y;
4 }
5
6 int main() {
7     int a = 8, b = 3;
8     int s = 0, d = 0;
9     make(a, b, &s, &d);
10    cout << s << " " << d;
11    return 0;
12 }
```

21. 阅读下列程序，写出输出结果：

```
1 void f1(int a[]) { a[0] += 1; }
2 void f2(int a[]) { a[1] = a[0] + a[1]; }
```

```

3 void f3(int a[]) { a[0] = a[1] - a[0]; }
4
5 int main() {
6     int a[2] = {3, 4};
7     f1(a);
8     f2(a);
9     f3(a);
10    cout << a[0] << " " << a[1];
11    return 0;
12 }
```

## 五、程序填空题

1. 完善下面的代码：使其输出一个空心等腰直角三角形，输出结果如下所示：

```

*
* *
*   *
*****
1 void show()
2 {
3     for (int i = 0; -----; i++)
4     {
5         for (int j = 0; -----; j++)
6             if ( ----- )
7                 cout << "*";
8             else
9                 -----;
10            -----;
11     }
12     return;
13 }
```

2. 完善下面的代码：在区间  $[a, b)$  内随机生成  $n$  个整数，判断其中哪些是完全平方数，并输出其算术平方根。

```

1 #include <iostream>
2 #include <cmath>
```

```
3 -----
4 -----
5
6 using namespace std;
7
8 -----
9
10 int main()
11 {
12     -----
13
14     cout << "输入正整数[a,b)范围和个数n: " << endl;
15     -----
16
17     if ( ----- )
18     {
19         cout << "输入不符合要求！应满足0<=a<b且n>0！" << endl;
20         return 0;
21     }
22
23     -----
24
25     for (int i = 0; i < n; i++)
26     {
27         int x = ----- ; // 随机生成[a,b)内的一个整数
28         double y = 0;
29
30         if (is_perfect_square(x, &y))
31         {
32             cout << "找到第" << ++cnt << "个完全平方数: " << x
33             << endl;
34             cout << "算术平方根为: " << y << endl;
35         }
36     }
37     return 0;
38 }
39 // 函数 is_perfect_square 判断整数 x 是否是完全平方数
40 // 并通过指针参数获得算术平方根
```

```

41 bool is_perfect_square( ----- )
42 {
43     for (int i = 2; i * i <= x; i++)
44         if ( ----- )
45         {
46             *root = i;
47             return true;
48         }
49
50     *root = ----- ;
51     return false;
52 }
```

## 六、函数编写题

1. 编写函数 `adjust`, 给定两个整型变量的地址 `p` 和 `q`, 以及一个整型阈值 `limit`, 按以下规则调整变量:

- 若  $*p + *q > limit$ , 交换  $*p$  与  $*q$ ;
- 否则, 将较小的那个变量的值修改为  $limit -$  较大的那个值。

函数无返回值, 必须通过指针参数修改实参。

```
1 void adjust(int* p, int* q, int limit);
```

2. 编写函数 `count_range`, 统计数组 `a` 中落在闭区间  $[L, R]$  内的元素个数, 并通过指针参数 `cnt` 返回结果。

```
1 void count_range(const int a[], int n, int L, int R, int* cnt);
```

3. 编写函数 `minmax_pos`, 找出数组 `a` 中最小值与最大值的下标, 并分别通过指针参数 `minPos`、`maxPos` 返回。若 `n <= 0`, 函数返回 `false`; 否则返回 `true`。

```
1 bool minmax_pos(const int a[], int n, int* minPos, int* maxPos);
```