

计算机科学基础 I——Homework 11

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-hw11；
- **截止时间：2026 年 1 月 9 日**

代码阅读题

1. 阅读下列程序，写出输出结果：

```
1  int a[2][3] = {{1,2,3},{4,5,6}};  
2  int *p = &a[0][0];  
3  cout << p[3] << endl;
```

2. 阅读下列程序，写出输出结果：

```
1  int a[3][4] = {  
2      {1, 2, 3, 4},  
3      {5, 6, 7, 8},  
4      {9, 10, 11, 12}  
5  };  
6  
7  int *p = &a[0][0];  
8
```

```

9 cout << a[1][3] << " ";
10 cout << p[1*4 + 3] << endl;

```

3. 阅读下列程序，写出输出结果：

```

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

```

4. 阅读下列程序，写出输出结果：

```

1 void func(int a[]) {
2     cout << "a[0]=" << a[0] << ", a[1]=" << a[1] << endl;
3 }
4 int main() {
5     int arr[5] = {10, 20, 30, 40, 50};
6     func(arr);
7     func(arr + 1);
8     func(arr + 2);
9     return 0;
10 }

```

5. 阅读下列程序，写出输出结果：

```

1 int main() {
2     char s[] = "hello";
3     cout << sizeof(s) << " " << strlen(s) << endl;
4 }

```

6. 阅读下列程序，写出输出结果：

```

1 char a[] = "ab\0cd";
2 char *b = a + 3;
3 cout << a << endl;

```

```

4 cout << strlen(a) << " " << sizeof(a) << endl;
5 cout << strlen(b) << endl;

```

7. 阅读下列程序，写出输出结果：

```

1 const char* p = "ABCDE";
2 cout << *(p + 1) << " " << (p + 2) << endl;

```

8. 阅读下列程序，写出输出结果：

```

1 char a[20] = "ab";
2 char b[20] = "cd";
3 strcat(a, b)
4 cout << a << endl;
5 cout << strlen(a) << endl;
6 cout << sizeof(a) << endl;

```

9. 阅读下列程序，写出输出结果：

```

1 char a[20] = "abde";
2 char b[20] = "cd";
3 strcpy(a, b);
4 cout << a << endl;

```

10. 阅读下列程序，写出输出结果：

```

1 char s1[] = "abc";
2 char s2[] = "abc";
3 if (strcmp(s1, s2)){
4     cout << "T" << endl;
5 }
6 else{
7     cout << "F" << endl;
8 }

```

11. 阅读下列程序，写出输出结果：

```

1 char a[20] = "abc";
2 char* p = a + 1;
3 cout << strlen(p) << " ";
4 cout << strcmp(p, "bc") << " ";
5 strcat(p, "!");
6 cout << a << endl;

```

12. 阅读下列程序，写出输出结果：

```
1 char* endp(char* s) {  
2     while (*s) ++s;  
3     return s;  
4 }  
5 int main() {  
6     char a[] = "hello";  
7     cout << (endp(a) - a) << endl;  
8 }
```

13. 阅读下列程序，写出输出结果：

```
1 void cutin(char* s) { s[2] = '\0'; }  
2 int main() {  
3     char a[] = "abcdef";  
4     cutin(a);  
5     cout << a << " " << (a + 3) << endl;  
6 }
```

14. 阅读下列程序，写出输出结果：

```
1 char* func(char* s) {  
2     char* p = s;  
3     while (*p) {  
4         if (*p >= 'a' && *p <= 'z')  
5             *p = *p - 'a' + 'A';  
6         p++;  
7     }  
8     return s;  
9 }  
10 int main() {  
11     char a[20] = "hello";  
12     cout << func(a) << endl;  
13 }
```

15. 阅读下列程序，写出输出结果：

```
1 char* func(char* s) {  
2     char* p = s;  
3     while (*p) {  
4         if (*p >= 'A' && *p <= 'Z')
```

```

5         *p = *p - 'A' + 'a';
6         p++;
7     }
8     return s;
9 }
10 int main() {
11     char a[20] = "AbC";
12     cout << func(a); << endl;
13 }

```

16. 阅读下列程序，写出输出结果：

```

1 char* func(char* dest, const char* src) {
2     char* p = dest;
3     while (*p) p++;
4     while (*src) {
5         *p = *src;
6         p++;
7         src++;
8     }
9     *p = '\0';
10    return dest;
11 }
12 int main() {
13     char a[20] = "abc";
14     func(a, "DEF");
15     cout << a << endl;
16 }

```

17. 阅读下列程序，写出输出结果：

```

1 char* func(char* dest, const char* src) {
2     char* p = dest;
3     while (*src) {
4         *p = *src;
5         p++;
6         src++;
7     }
8     *p = '\0';
9     return dest;
10 }

```

```

11 int main() {
12     char a[20] = "abcdef";
13     char b[] = "XY";
14     my_strcpy(a + 2, b);
15     cout << a << endl;
16 }

```

18. 阅读下列程序，写出输出结果：

```

1 char a[][6] = {"cat", "car", "dog", "do"};
2 int i = 0;
3 while (i < 4 && strcmp(a[i], "dog") != 0) ++i;
4 cout << i << " " << strlen(a[3]) << endl;

```

19. 阅读下列程序，写出输出结果：

```

1 int main() {
2     char a[][6] = {"cat", "car", "dog", "do"};
3     cout << strlen(a[0]) << " " << strlen(a[3]) << endl;
4 }

```

20. 阅读下列程序，写出输出结果：

```

1 const char* w[] = {"one", "two", "three"};
2 cout << w[1] << " " << (w[2] + 2) << endl;

```

21. 阅读下列程序，写出输出结果：

```

1 char a[] = "abcd";
2 char b[] = "XY";
3 char c[] = "pqrs";
4 char* p[] = {a, b, c};
5
6 cout << *p[0] << " ";
7 cout << *(p[0]++) << " ";
8 cout << *p[0] << " ";
9 cout << a << endl;

```

22. 阅读下列程序，写出输出结果：

```

1 char s1[] = "az";
2 char s2[] = "BY";

```

```

3 char s3[] = "mn";
4 char* p[] = {s1, s2, s3};
5
6 cout << (*p[1])++ << " ";
7 cout << s2 << " ";
8 cout << *++p[1] << " ";
9 cout << s2 << endl;

```

23. 阅读下列程序，写出输出结果：

```

1 int a = 1, b = 100;
2 int& r = a;
3 r = b;
4 b = 200;
5 cout << a << " " << b << " " << r << endl;

```

24. 阅读下列程序，写出输出结果：

```

1 int a = 5, b = 9;
2 int& r = a;
3 r += 3;
4 r = b;
5 b += 1;
6 cout << a << " " << b << " " << r << endl;

```

25. 阅读下列程序，写出输出结果：

```

1 int a = 4;
2 const int& cr = a;
3 a += 3;
4 cout << a << " " << cr << endl;

```

26. 阅读下列程序，写出输出结果：

```

1 int a = 10;
2 const int& cr = a;
3 cr = 30;
4 cout << cr << endl;

```

27. 阅读下列程序，写出输出结果：

```

1 void inc(int& x) { x += 1; }
2
3 int main() {
4     double a = 10;
5     inc(a);
6     cout << a << endl;
7 }

```

28. 阅读下列程序，写出输出结果：

```

1 double inc(double& x) { x += 2; return x;}
2 int inc(int x) { x += 1; return x;}
3
4 int main() {
5     cout << inc(1.0) << endl;
6 }

```

29. 阅读下列程序，写出输出结果：

```

1 void add1(int& x) { x += 1; }
2
3 int main() {
4     int a = 10;
5     int& r = a;
6     add1(r);
7     add1(a);
8     cout << a << " " << r << endl;
9 }

```

30. 阅读下列程序，写出输出结果：

```

1 void swap2(int& x, int& y) {
2     int t = x; x = y; y = t;
3 }
4
5 int main() {
6     int a = 3;
7     int& r = a;
8     swap2(a, r);
9     cout << a << endl;
10 }

```


31. 阅读下列程序，写出输出结果：

```
1 void inc(int x) { x += 10; }
2 void inc2(int& x) { x += 10; }
3
4 int main() {
5     int a = 1;
6     int& r = a;
7     inc(r);
8     inc2(r);
9     cout << a << endl;
10 }
```

代码填空题

1. 补全下面的程序，实现如下功能：

程序运行时，先使用当前时间设置随机数种子，然后在区间 $[0, 9]$ 内随机生成 10 个整数，统计每个数字出现的次数，最后按顺序输出数字 $0 \sim 9$ 的出现次数。（说明：随机数生成是考试重点，必须要掌握如何生成指定范围的随机数）

例如，若随机生成的序列为：

2 5 2 9 0 5 2 1 9 2

则输出结果应为：

0:1 1:1 2:4 3:0 4:0 5:2 6:0 7:0 8:0 9:2

```
1 #include <iostream>
2 -----
3 -----
4 using namespace std;
5
6 int main() {
7     int cnt[10] = {0};
8
9     -----;
10
11     for (int i = 0; i < 10; ++i) {
12         int x = -----;
```

```

13         -----;
14     }
15
16     for (int i = 0; i < 10; ++i) {
17         cout << i << ":" << cnt[i] << " ";
18     }
19     cout << endl;
20
21     return 0;
22 }

```

函数编写题

1. 编写函数 `sumRow`，使用**一级指针**访问并操作一个二维整型数组，计算并返回指定行中所有元素的和。

已知有一个二维数组，逻辑上为 `row` 行、`col` 列，在调用函数时，二维数组的首地址被当作**一级指针**传入。

函数原型如下：

```
1 int sumRow(int* a, int row, int col, int r);
```

其中：

- `a` 指向二维数组的首元素（即 `&a[0][0]`）；
- `row` 表示数组的行数；
- `col` 表示数组的列数；
- `r` 表示要计算的行号（从 0 开始编号， $0 \leq r < row$ ）。

函数需要返回第 `r` 行所有元素之和。

要求：

- 必须通过一级指针 `a` 进行访问；
- 不允许使用二维数组下标形式（如 `a[i][j]`）；

调用示例（仅供理解）：

```

1 int main() {
2     int b[3][4] = {
3         {1, 2, 3, 4},
4         {5, 6, 7, 8},
5         {9, 10, 11, 12}
6     };
7     cout << sumRow(&b[0][0], 3, 4, 1) << endl; // 输出 26
8 }

```

2. 编写函数 `isEqual`，用于比较两个以 ‘\0’ 结尾的 C 风格字符串是否完全相等。

若两个字符串在长度和对应位置的字符均相同，则认为它们相等；否则认为它们不相等。

函数原型如下：

```

1 bool isEqual(const char* s1, const char* s2);

```

- 若两个字符串相等，函数返回 `true`；
- 若两个字符串不相等，函数返回 `false`。

不允许使用 `string` 类型，不允许使用任何字符串比较相关的库函数（如 `strcmp`）。

3. 编写函数 `insertStr`，实现将一个字符串插入到另一个字符串的指定位置。

给定一个以 ‘\0’ 结尾的 C 风格字符串 `s`，以及另一个以 ‘\0’ 结尾的字符串 `t`，和一个非负整数位置 `pos`，请将字符串 `t` 插入到字符串 `s` 的第 `pos` 个字符之后。

插入完成后，原字符串 `s` 中从位置 `pos` 开始的字符整体向后移动，字符串 `t` 的内容被完整插入其中。

例如：

- `s = "abcdef", t = "XYZ", pos = 2` 插入后，`s` 变为 `"abcXYZdef"`

函数原型如下：

```

1 void insertStr(char* s, const char* t, int pos);

```

说明与要求：

- 保证字符串 `s` 所在的字符数组空间足够大，能够容纳插入后的结果；
- 假设 $0 \leq pos < strlen(s)$ ；
- 插入操作需要在原字符串 `s` 上完成；