

2023 年 6 月 GESP C++ 三级真题解析

CCF 编程能力等级认证，英文名 Grade Examination of Software Programming（以下简称 GESP），由中国计算机学会发起并主办，是为青少年计算机和编程学习者提供学业能力验证的平台。GESP 覆盖中小学全学段，符合条件的青少年均可参加认证。GESP 旨在提升青少年计算机和编程教育水平，推广和普及青少年计算机和编程教育。

GESP 考察语言为图形化（Scratch）编程、Python 编程及 C++编程，主要考察学生掌握相关编程知识和操作能力，熟悉编程各项基础知识和理论框架，通过设定不同等级的考试目标，让学生具备编程从简单的程序到复杂程序设计的编程能力，为后期专业化编程学习打下良好基础。

本次为大家带来的是 2023 年 6 月份，C++ 三级考试真题解析。

一、单选题（每题 2 分，共 30 分）

1. 高级语言编写的程序需要经过以下（ ）操作，可以生成在计算机上运行的可执行代码。

- A. 编辑
- B. 保存
- C. 调试
- D. 编译

【答案】D

【考纲知识点】编程环境(一级)

【解析】本题属于考察计算机基础知识中的编辑、编译、解释、调试的概念；其中编辑是编写修改代码，保存是将代码保存下来，调试是测试运行代码，而编译是将源程序翻译成可执行代码，所以本题正确答案为 D。

2. 二进制数 11.01 在十进制下是（ ）。

- A. 3.01
- B. 3.05

C. 3.125

D. 3.25

【答案】D

【考纲知识点】进制转换 (三级)

【解析】本题属于考察计算机基础知识中的进制与进制转换；二进制 11.01 换算成 10 进制为 $1 * 2^1 + 1 * 2^0 + 0 * \frac{1}{2} + 1 * \frac{1}{2^2} = 3.25$ 。所以本题正确答案为 D。

3. 已知大写字符'A'的 ASCII 编码的十六进制表示为 0x41，则字符'F'的 ASCII 编码的十六进制表示为 ()。

A. 46

B. 47

C. 48

D. 49

【答案】A

【考纲知识点】ASCII 编码 (二级)

【解析】本题属于考察计算机基础知识中的 ASCII 码；已知'A'的 ASCII 编码十六进制表示为 0x41，则'F'的 ASCII 编码比'A'的 ASCII 编码大 5，所以字符'F'的 ASCII 编码十六进制表示为 46。所以本题正确答案为 A。

4. 以下哪个不是 C++语言中的运算符？ ()

A. &

B. &&

C. *

D. **

【答案】D

【考纲知识点】基本运算 (一级)，位运算 (三级)

【解析】本题属于考察计算机基础知识中的运算符，其中&、&&、*分别可以表示按位与、逻辑与和乘法，但**不是 C++语言中的运算符。本题正确答案为 D。

5. 如果字符串定义为 `char str[] = "Hello";`，则字符数组 `str` 的长度为（ ）。

- A. 0
- B. 5
- C. 6
- D. 7

【答案】C

【考纲知识点】字符串（三级）

【解析】本题属于考察计算机基础知识中的字符串概念。数组初始化时[]内可省略数组长度，此时数组长度为初始值的长度。字符串"Hello"的长度为5，但是字符串最后需要额外保存一个'\0'符号作为结束位，所以字符数组 `str` 的长度为6。所以本题正确答案为C。

6. 一个数组定义为 `double array[3];`，则这个数组占用内存的大小为（ ）。

- A. 24
- B. 12
- C. 6
- D. 3

【答案】A

【考纲知识点】计算机的存储（二级），一维数组（三级）

【解析】本题属于考察计算机基础知识中的内存概念，`double` 类型的数据占用内存为8字节，`array` 数组共使用了3个`double` 类型的数据，所以内存大小为24。所以本题正确答案为A。

7. 以下数组定义，符合C++语言语法的是（ ）。

- A. `int a[];`
- B. `int b['3'];`
- C. `int c[3.0];`
- D. `int[3] d;`

【答案】B

【考纲知识点】一维数组（三级）

【解析】本题属于考察计算机基础知识中的数组概念，在 C++ 中每个字符都有对应的 ASCII 编码，当需要使用整数类型的值时，可以将字符通过对应的 ASCII 编码转换为整数，所以 `int b['3'];` 等价于 `int b[51];`。所以本题正确答案为 B。

8. 下列关于进制的叙述，不正确的是（ ）。

- A. 正整数的二进制表示中只会出现 0 和 1。
- B. 10 不是 2 的整数次幂，所以十进制数无法转换为二进制数。
- C. 从二进制转换为 8 进制时，可以很方便地由低到高将每 3 位二进制位转换为对应的一位 8 进制位。
- D. 从二进制转换为 16 进制时，可以很方便地由低到高将每 4 位二进制位转换为对应的一位 16 进制位。

【答案】B

【考纲知识点】进制转换（三级）

【解析】本题属于考察计算机基础知识中的进制与进制转换概念，任意数字都可以在不同的进制之间转换，所以本题正确答案为 B。

9. 下列关于 C++ 语言中数组的叙述，不正确的是（ ）。

- A. 数组必须先定义后使用。
- B. 数组的所有元素在内存中是连续存放的。
- C. 除了字符数组，在定义数组时“[]”内必须有常数。
- D. 不能对数组赋值，但可以对数组的每个基础类型的元素赋值。

【答案】C

【考纲知识点】一维数组（三级）

【解析】本题属于考察计算机基础知识中的数组概念，任意类型的数组，只要有初始化，在数组定义时就可以省略[]内的常数。所以本题正确答案为 C。

10. 一个 int 类型的值，做以下哪个操作，一定会变回原来的值？（ ）

- A. 左移 3 位，再右移 3 位。

- B. 右移 3 位，再左移 3 位。
- C. 按位或 7，再按位与-8。
- D. 按位异或 7，再按位异或 7。

【答案】D

【考纲知识点】位运算（三级）

【解析】本题属于考察计算机基础知识中的位运算相关概念，A 选项可能存在越界的问题，B 选项会将低位的 3 位清零，C 选项肯定是错的，D 选项异或两次 7 等同于原数，所以本题正确答案为 D。

11. 如果 a 和 b 均为 int 类型的变量，下列表达式能正确判断“a 等于 b”的是()。

- A. $((a / b) == 1)$
- B. $((a \& b) == a)$
- C. $((a \wedge b) == 0)$
- D. $((a | b) == b)$

【答案】C

【考纲知识点】位运算（三级）

【解析】本题属于考察计算机基础知识中的位运算相关概念。

选项 A，例如 a 为 5，b 为 3，则 a/b 结果为 1，但 a 并不等于 b；

选项 B，例如 a 为 5，b 为 15，则 a&b 结果为 5，但 a 并不等于 b；

选项 C， $(a \wedge b) == 0$ 说明 a 和 b 的二进制位上的每一位都相等，等价于“a 等于 b”；

选项 D，例如 a 为 5，b 为 15，则 a|b 结果为 15，但 a 并不等于 b；

所以本题正确答案为 C。

12. 如果 a 为 int 类型的变量，下列哪个表达式可以正确求出满足“小于等于 a 且是 4 的倍数”的整数中最大的？()

- A. $(a \& (\sim 3))$
- B. $((a \ll 2) \gg 2)$
- C. $(a \wedge 3)$
- D. $((a - 1) | 3) + 1$

【答案】A

【考纲知识点】位运算（三级）

【解析】本题属于考察计算机基础知识中的位运算相关概念， $a \&$ 一个数得到的必然是小于等于 a 的数，同时题目要求是 4 的倍数，也就是最后两位要等于 0， (~ 3) 正好是除了最后两位是 0，其余全是 1，满足要求，所以本题正确答案为 A。

13. 在下列代码的横线处填写（ ），可以使得输出是“24 12”。

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int a = 12, b = 24;
5      _____; // 在此处填入代码
6      a = a ^ b;
7      b = a ^ b;
8      cout << a << " " << b << endl;
9      return 0;
10 }
```

- A. $a = a \wedge b$
- B. $b = a \wedge b$
- C. $a = a + b$
- D. $b = a + b$

【答案】B

【考纲知识点】位运算（三级）

【解析】本题属于考察计算机基础知识中的位运算相关概念，因为异或运算中 $a \wedge a = 0$; $a \wedge 0 = a$ ，当选项为 B 时，第 5 行 $b' = (a \wedge b)$ ，第 6 行 $a' = a \wedge b' = a \wedge (a \wedge b) = b$ ，第 7 行 $b'' = a \wedge b' = b \wedge (a \wedge b) = a$ ，正好对 a 、 b 进行了交换。所以本题正确答案为 B。

14. 在下列代码的横线处填写（ ），可以使得输出是“2”。

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int array[5] = {3, 7, 5, 2, 4};
5      int min = 0;
6      for (int i = 0; i < 5; i++)
7          if (_____) // 在此处填写代码
8              min = array[i];
9      cout << min << endl;
10     return 0;
11 }
```

- A. $\text{min} > \text{array}[i]$
- B. $\text{min} < \text{array}[i]$
- C. $\text{min} = \text{array}[i]$
- D. 以上均不对。

【答案】 D

【考纲知识点】 循环结构(一级),一维数组 (三级)

【解析】 本题属于考察计算机基础知识中的数组相关概念，从结果输出 2 以及变量名字 min 上可以看出，题目本身想求一个最小值，但是初始化为 $\text{min}=0$ ，所以选项 ABC 的最终结果都不会等于 2，所以本题正确答案为 D。

15. 在下列代码的横线处填写 ()，可以使得输出不是“31”。

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      int array[5] = {1, 2, 4, 8, 16};
5      int res = 0;
6      for (int i = 0; i < 5; i++)
7          _____; // 在此处填写代码
8      cout << res << endl;
9      return 0;
10 }
```

- A. $\text{res} = \text{res} + \text{array}[i]$
- B. $\text{res} = \text{res} \& \text{array}[i]$
- C. $\text{res} = \text{res} | \text{array}[i]$

D. $res = res \wedge array[i]$

【答案】 B

【考纲知识点】 位运算(三级)

【解析】 本题属于考察计算机基础知识中的数组和位运算相关概念，结果输出 31 的二进制位是 11111，而数组中的 5 个数的二进制分别是 1、10、100、1000 和 10000，所以选项 A、C、D 都会输出 31，只有 B 选项会输出 0，所以本题正确答案为 B。

二、判断题 (每题 2 分, 共 20 分)

1. 一个算法可以用不同的形式来描述，但要求描述比较规范，因此不能用自然语言描述。

【答案】 ×

【考纲知识点】 算法的概念与描述 (三级)

【解析】 本题属于考察算法相关概念，算法不仅可以用程序语言描述，也可以通过自然语言描述，例如中文等，所以本题错误。

2. 域名是由一串用点分隔的名字来标识互联网上一个计算机或计算机组的名称，CCF 编程能力等级认证官方网站的域名是 `gesp.ccf.org.cn`，其中顶级域名是 `gesp`。

【答案】 ×

【考纲知识点】 计算机存储与网络 (二级)

【解析】 本题属于考察域名相关概念，域名是由两个或两个以上的词构成，中间用点号分隔开，最右边的那个词称为顶级域名，所以顶级域名是 `cn`，所以本题错误。

3. 数据编码方式只有原码、反码、补码三种。

【答案】 ×

【考纲知识点】 数据编码(三级)

【解析】 本题属于考察数据编码相关概念，数据编码的方式非常多，例如格雷码、哈夫曼编码等，它们适合使用的场景各不相同。所以本题错误。

4. 在 C++ 语言中，长度为 n 的数组，合理的下标范围是从 0 到 n ，包括 0 和 n 。

【答案】 ×

【考纲知识点】一维数组(三级)

【解析】本题属于考察数组相关概念，长度为 n 的数组，合理的下标范围是 0 到 $n-1$ ，不包括 n ，所以本题错误。

5. 字符常量 `'\0'` 常用来表示字符串结束，它和字符常量 `'0'` 是不同的。

【答案】√

【考纲知识点】字符串(三级)

【解析】本题属于考察字符串相关概念，字符常量 `'\0'` 是字符串和字符数组的结束标记，它的 ASCII 码为 0 ，字符常量 `'0'` 的 ASCII 码为 48 ，与 `'\0'` 是不同的字符常量。所以本题正确。

6. 在 C++ 语言中，可以使用字符（如 `'0'`）作为数组下标。

【答案】√

【考纲知识点】位运算(三级) **【解析】**本题属于考察数组相关概念，在 C++ 中每个字符都有对应的 ASCII 编码，当需要使用整数类型的值时，可以将字符通过对应的 ASCII 编码转换为整数，`'0'` 的 ASCII 编码为 48 ，所以 `int b['0']` 等价于 `int b[48]`，所以本题正确。

7. 在 C++ 语言中，数组被定义时，它的大小就确定了。

【答案】√

【考纲知识点】一维数组(三级)

【解析】本题属于考察数组相关概念，在 C++ 中定义数组的时候就需要有明确的数组大小，所以本题正确。

8. 计算机中存储的数据都是二进制形式。因此，在使用 C++ 语言编写程序时，将所有十进制数改写为相同数值的二进制数，会使得程序运行效率更高。

【答案】×

【考纲知识点】编程环境(一级)

【解析】本题属于考察程序的基本概念，无论编写程序时使用十进制还是二进制，编译器都会翻译为二进制机器指令运行，所以运行效率是一样的，所以本题错误。

9. 在 C++ 语言中，表达式 `(0xf == 015)` 的值为 `true`。

【答案】×

【考纲知识点】进制转换(三级)

【解析】本题属于考察程序的进制相关概念，0xf 是 16 进制下的 f，等于 10 进制的 15；015 是 8 进制下的 15，等于 10 进制的 13。可见它们不相等，所以本题错误。

10. 如果 a 为 int 类型的变量，且表达式((a | 3) == 3)的值为 true，则说明 a 在从 0 到 3 之间（可能为 0、可能为 3）。

【答案】√

【考纲知识点】位运算(三级)

【解析】本题属于考察位运算相关概念，((a|3)==3)说明 a 只在二进制的最低位和第二低位可能为 1，所以 $0 \leq a \leq 3$ ，所以本题正确。

三、编程题 (每题 25 分, 共 50 分)

1. 春游

【问题描述】

老师带领同学们春游。已知班上有 N 位同学，每位同学有从 0 到 $N - 1$ 的唯一编号。到了集合时间，老师确认是否所有同学都到达了集合地点，就让同学们报出自己的编号。到达的同学都会报出的编号，不会报出别人的编号，但有的同学很顽皮，会多次报出。你能帮老师找出有哪些同学没有到达吗？

【输入描述】

输入包含 2 行。第一行包含两个整数 N 和 M ，表示班级有 N 位同学，同学们共有 M 次报出编号。约定 $2 \leq N, M \leq 1000$ 。

第二行包含 M 个整数，分别为 M 次报出的编号。约定所有编号都在合理范围内。

【输出描述】

输出一行。如果所有同学都到达，则输出 N ；否则由小到大输出所有未到达的同学编号，空格分隔。

【样例输入 1】

```
3 3
0 2 1
```

【样例输出 1】

```
3
```

【样例输入 2】

```
3 5
0 0 0 0 0
```

【样例输出 2】

```
1 2
```

【解题思路】

1. 使用 `arrive` 数组来记录每个同学是否到达，初始为 `false`，表示没有到达。
2. 每有一个同学报出编号，就将 `arrive` 数组对应的编号改为 `true`。
3. 遍历 `arrive` 数组，将数组中对应位置为 `false` 的下标输出并特判所有人均到达的情况。

【考纲知识点】 循环结构（一级），模拟法、一维数组（三级）

【参考程序】

```
#include <iostream>
using namespace std;
bool arrive[1000];
int main() {
    int n = 0, m = 0;
    cin >> n >> m;
    // 初始化 arrive 数组为所有同学均未报到
    for (int i = 0; i < n; i++)
        arrive[i] = false;
    // 依次报到 m 次
    for (int i = 0; i < m; i++) {
        int code = 0;
        cin >> code;
        arrive[code] = true;
    }
    // 依次检查 n 位同学是否到达
    bool all = true;
    for (int i = 0; i < n; i++) {
        if (!arrive[i]) {
            if (all) {
                cout << i;
                all = false;
            } else {
                cout << " " << i;
            }
        }
    }
    // 处理全部到达的特殊情况
    if (all)
        cout << n;
    cout << endl;
    return 0;
}
```

2. 密码合规检测

【问题描述】

网站注册需要有用户名和密码，编写程序以检查用户输入密码的有效性。合规的密码应满足以下要求：

1、只能由 a-z 之间 26 个小写字母、A-Z 之间 26 个大写字母、0-9 之间 10 个数字以及!@#\$四个特殊字符构成。

2、密码最短长度：6 个字符，密码最大长度：12 个字符。

3、大写字母、小写字母和数字必须至少有其中两种，以及至少有四个特殊字符中的一个。

【输入描述】

输入一行不含空格的字符串。约定长度不超过 100。该字符串被英文逗号分隔为多段，作为多组被检测密码。

【输出描述】

输出若干行，每行输出一组合规的密码。

输出顺序以输入先后为序，即先输入则先输出。

【样例输入 1】

```
seHJ12!@,sjdkffH$123,sdf!@&12HDHa!,123&^YUhg@!
```

【样例输出 1】

```
seHJ12!@  
sjdkffH$123
```

【样例解释 1】

输入被英文逗号分为了四组被检测密码：“seHJ12!@”、“sjdkffH\$123”、“sdf!@&12HDHa!”、“123&^YUhg@!”。其中，“sdf!@&12HDHa!”长度超过 12 个字符，不合规；“123&^YUhg@!”包含四个特殊字符之外的字符“^”，不合规。

【解题思路】

1. 首先遍历输入的字符串 `line` 并按照','进行分隔。
2. 对于分隔出来的每一个密码，先判断密码长度是否符合要求，然后遍历所有的字符，并用 `hasC`，`hasL`，`hasD`，`hasS` 分别记录是否存在大写字母，小写字母，数字以及特殊字符，若存在以上四种字符外的其它字符则直接返回非法。

3. 判断是否存在特殊字符，若不存在返回非法。
4. 判断是否存在两种及以上的大写字母、小写字母和数字，若不存在返回非法。
5. 以上情况都存在，返回合法并输出。

【考纲知识点】多层分支/循环结构（二级），模拟法、字符串（三级）

【参考程序】

```
#include <iostream>
using namespace std;
char line[101];
char pwd[101];
// 检查从 str 开始、长度为 l 的密码是否合规
bool check(char * str, int l) {
    if (l < 6 || l > 12)
        return false;
    bool hasC = false, hasL = false, hasD = false, hasS = false;
    for (int i = 0; str[i] != '\0'; i++) {
        if ('A' <= str[i] && str[i] <= 'Z') {
            hasC = true;
        } else if ('a' <= str[i] && str[i] <= 'z') {
            hasL = true;
        } else if ('0' <= str[i] && str[i] <= '9') {
            hasD = true;
        } else if (str[i] == '!' || str[i] == '@' ||
            str[i] == '#' || str[i] == '$') {
            hasS = true;
        } else
            return false;
    }
    if (!hasS)
        return false;
    if (hasC + hasL + hasD < 2)
        return false;
    return true;
}
int main() {
    cin >> line;
    // 按逗号对输入进行切分，并依次判断
    int len = 0;
    for (int i = 0; line[i] != '\0'; i++) {
        if (line[i] != ',') {
            pwd[len] = line[i];
        }
    }
}
```

```
        len++;
    } else {
        pwd[len] = '\\0';
        if (check(pwd, len))
            cout << pwd << endl;
        len = 0;
    }
}
if (len > 0) {
    pwd[len] = '\\0';
    if (check(pwd, len))
        cout << pwd << endl;
}
return 0;
}
```