

GESP C++ 三级模拟试题 2

一、 选择题

1. (样题) 下列关于负数的原码、反码、补码的描述中, 正确的是 ()。

- A. 原码和反码互为按位取反(符号位除外), 补码为反码加 1
- B. 原码和反码互为按位取反(符号位除外), 补码为原码加 1
- C. 反码和补码互为按位取反(符号位除外), 原码为反码加 1
- D. 补码和原码互为按位取反(符号位除外), 反码为补码加 1

【答案】A

【考纲知识点】数据编码

【解析】原码：最高位是符号位，0代表正数，1代表负数，非符号位为该数字绝对值的二进制。反码：正数的反码与原码一致，负数的反码是对原码按位取反，只是最高位(符号位)不变。补码：正数的补码与原码一致，负数的补码是对原码按位取反加1，符号位不变。所以选A。

2. (2023年6月) 已知大写字母'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。

3. (2023年6月) 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】变基本运算(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。

4. (2023年6月) 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。

5. （2023年6月）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。

6. （2023年9月）以下数组定义，符合C++语言语法的是（ ）。
- A. `double a[];`
 - B. `double b[] = {1, 2.0, '3'};`
 - C. `double c[3.0];`
 - D. `double[] d = new double[3];`

【答案】B

【考纲知识点】数组知识

【解析】本题属于考察C++语言的数组知识，A的写法需要明确数组大小；C中数组大小必须是整数；D需要用指针类型指向一个new的数组。选B。

7. （样题）对于一个十进制数 37，以下哪个是它的二进制表示（ ）。
- A. 10101
 - B. 100101
 - C. 101001
 - D. 1000101

【答案】B

【考纲知识点】进制

【解析】 $37=2^5+2^2+1^2+0$ ，所以就是100101，所以选B。

8. （2023年9月）下列关于C++语言中数组的叙述，不正确的是（ ）。

- A. 可以定义 0 个元素的数组。
- B. 不能定义-1 个元素的数组。
- C. 数组下标越界访问会产生编译错误。
- D. 程序运行时发生数组下标的越界访问，程序依然可能正常结束。

【答案】C

【考纲知识点】数组知识

【解析】本题属于考察C++语言的数组知识，选择不正确的，在不同编译器下，数组越界不一定会产生编译错误，选C。

9. (样题) 通常用下列哪种方式来描述算法? ()。

- A. 汇编语言
- B. 伪代码
- C. SQL
- D. CSS

【答案】B

【考纲知识点】算法

【解析】汇编语言是一种低级语言，使用简单的助记符号来表示计算机指令和操作码，直接对应底层的机器指令。伪代码是一种高级语言，使用类似自然语言的语法来描述算法和程序流程，不依赖于具体的底层硬件和指令集。所以选B。

10. (2023年6月) 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. (2023年6月) 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. $((a < 2) > 2)$
- C. $(a \wedge 3)$
- D. $((a - 1) \mid 3) + 1$

【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， $a \& \text{一个数}$ 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。

12. (2023年9月) 在下列代码的横线处填写 ()，可以使得输出是 "20 10"。

```
#include <iostream>
using namespace std;
int main(){
    int a=10,b=20;
    a=(a<<8)|b;
    _____//在此处填入代码
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

- A. a = a >> 8; b = a & 0xff
- B. b = a >> 8; a = a & 0xff;
- C. a = b; b = a & 0xff;
- D. b = a; a = b

【答案】B

【考纲知识点】基本运算

【解析】本题属于考察 C++ 语言的运算知识，程序的目的是实现 a 和 b 的数据交换，用 a 的低 8 位保留 b 的值，原来的数值保存在 a 的高 8 位，因此 $b = a >> 8$ 即得到 a 原来的值。将 a 的高 8 位清空，与 0xff 做与运算即可，0xff 的高 8 位是 0，得到 b 之前的结果，选 B。

13. (2023年6月) 在下列代码的横线处填写 ()，可以使得输出是 "2"。

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

- A. min > array[i]
- B. min < array[i]
- C. min = 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 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。

14. (样题) 在下列代码的横线处填写 ()，可以使得输出是 "17 11"。

```
#include <iostream>
using namespace std;
int main(){
    int a=11, b=17;
    a=____; // 在此处填入代码
    b=a^b;
    a=a^b;
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

- A. a+b
- B. a-b
- C. a^b
- D. a&b

【答案】C

【考纲知识点】位运算

【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 C 时，第 5 行 $a' = (a \wedge b)$ ，第 6 行 $b' = a \wedge b' = a \wedge (a \wedge b) = b$ ，第 7 行 $a'' = a \wedge b' = b \wedge (a \wedge b) = a$ ，正好对 a、b 进行了交换。所以本题正确答案为 C。

15. (2023年9月) 在下列代码的输出是 ()。

```
#include <iostream>
using namespace std;
int main(){
    int array[10];
    for(int i=0; i<10; i++)
        array[i]=i;
    for(int p=2; p<10; p++)
        if(array[p]==p)
            for(int n=p; n<10; n+=p)
                array[n]=array[n]/p*(p-1);
    int res=0;
    for(int n=1; n<10; n++)
        res+=array[n];
    cout<<res<<endl;
    return 0;
}
```

- A. 15
- B. 28
- C. 45
- D. 55

【答案】B

【考纲知识点】多层循环结构，数组

【解析】本题属于考察C++语言的数组知识，array数组赋值分别是0到9，然后注

意里面的双重循环。p是从2到9，对array数组重新赋值。例如，p=2时，array[2]=2,条件成立，进入第2重循环，n的范围是2到9，注意步长是2，对每个数组元素重新赋值，数组重新变为0 1 1 3 2 5 3 7 4 9。一直到循环完成，累加array数组，结果是B。

二、 判断题

1. (2023年9月) 二进制数101.101在十进制下是5.005。
【答案】错误
【考纲知识点】 进制转换
【解析】本题是计算机二进制和转换十进制的知识，整数部分是5，小数部分是0.625。
2. (2023年6月) 数据编码方式只有原码、反码、补码三种。
【答案】错误
【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。
3. (2023年6月) 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】错误
【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。
4. (2023年9月) 在 C++语言中，定义数组时，[]中必须指定元素个数。
【答案】错误
【考纲知识点】 数组的定义
【解析】本题是计算机数组知识，数组定义方式有很多，正确的也包括：int d[]={1,2,3};这种格式。
5. (样题) 在 C++语言中，字符串是以'\0'结尾的字符数组。
【答案】正确
【考纲知识点】 字符数组
【解析】字符串是以'\0'结尾的字符数组，题目说法正确。
6. (2023年9月) 在C++语言中，所有int类型的值，经过若干次左移操作 (<<) 后，它们的值总会变为0。
【答案】正确
【考纲知识点】 位运算
【解析】本题是计算机位运算知识，左移后，后面的位数用0补充，所以移动若干次，都会变成0。
7. (样题) C++语言中数字的符号位是不参与位运算的。
【答案】错误

【考纲知识点】 位运算

【解析】对于有符号整数类型，进行位运算可能会导致符号位的变化。

8. (2023年9月) 著名的哥德巴赫猜想：任一大于2的偶数都可写成两个素数之和。我们可以通 过枚举法来证明它。

【答案】 错误

【考纲知识点】 枚举算法

【解析】本题是计算机算法知识，枚举法是枚举所有的可能，枚举不出所有的偶数，所以不能用枚举法证明哥德巴赫猜想。

9. (2023年6月) 在C++语言中，表达式(0xf == 015)的值为 true。

【答案】 错误

【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. (2023年9月) 如果a为int类型的变量，且表达式((a & 1) == 0) 的值为true，则说明a是偶数。

【答案】 正确

【考纲知识点】 位运算和比较运算符

【解析】本题是计算机位运算和比较运算符知识，a是整数，只有是偶数和1做与 运算的结果才等于0。

三、 编程题

1. 比例简化(NOIP2014普及组)

【问题描述】

在社交媒体上，经常会看到针对某一个观点同意与否的民意调查以及结果。例如，对某一观点表示支持的有 1498 人，反对的有 902 人，那么赞同与反对的比例可以简单的记为 1498:902。不过，如果把调查结果就以这种方式呈现出来，大多数人肯定不会满意。因为这个比例的数值太大，难以一眼看出它们的关系。对于上面这个例子，如果把比例记为 5:3，虽然与真实结果有一定的误差，但依然能够较为准确地反映调查结果，同时也显得比较直观。

现给出支持人数 A，反对人数 B，以及一个上限 L，请你将 A 比 B 化简为 A' 比 B'，要求在 A' 和 B' 均不大于 L 且 A' 和 B' 互质（两个整数的最大公约数是 1）的前提下， $\frac{A'}{B'} \geq \frac{A}{B}$ 且 $\frac{A'}{B'} - \frac{A}{B}$ 的值尽可能小。

【输入描述】

共一行，包含三个整数 A, B, L，每两个整数之间用一个空格隔开，分别表示支持人数、反对人数以及上限。

【输出描述】

共一行，包含两个整数 'A', B'，中间用一个空格隔开，表示化简后的比例。

【输入样例】

1498 902 10

【输出样例】

5 3

【考纲知识点】双层循环、枚举法

【解题思路】

1. 定义三个变量，输入三个整数
2. 使用循环，枚举所有可能的分子和分母
3. 查找满足条件的分子和分母，条件1可以转换成 $i*b \geq j*a$ ，条件2可以转换成 $\text{abs}(i*b-j*a) < \text{abs}(k1*b-k2*a)$ 。保证互质的关键是从小到大枚举i，从大到小枚举j，第一个符合条件的答案肯定就是最简分数。
4. 输出结果。

【参考代码】

```
#include <iostream>
using namespace std;
int main(){
    int a,b,l;
    cin>>a>>b>>l;
    int k1=1000000;
    int k2=1;
    for(int i=1;i<=l;i++){
        for(int j=l;j>=1;j--){
            if((i*b>=j*a)&&(abs(i*b-j*a)<abs(k1*b-k2*a))){
                k1=i;
                k2=j;
            }
        }
    }
    cout<<k1<<" "<<k2;
    return 0;
}
```

2. 密码合规检测(2023年6月)

【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】

```
3seHJ12!@,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）

【参考代码】

```

#include <iostream>
using namespace std;
bool check(string str){
    if(str.length()>12||str.length()<6)return false;
    // 标记是否含大写字母 小写字母 数字 特殊符号
    bool hasC=false,hasL=false,hasD=false,hasS=false;
    for(int i=0;i<str.length();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]=='$')
            hasS=true;
        else return false; // 含有其他字符, 不合法
    }
    // 判断大写字母、小写字母和数字必须至少有其中两种
    if(hasC+hasL+hasD<2)return false;
    // 判断以及至少有四个特殊字符!@#$中的一个
    if(!hasS)return false;
    return true;
}
int main(){
    string line;
    cin>>line;
    int pos=line.find(',');
    int begin=0;
    string pwd="";
    while(pos!=-1){
        for(int i=begin;i<pos;i++)pwd+=line[i];
        if(check(pwd))cout<<pwd<<endl;
        begin=pos+1;
        pos=line.find(',',begin);
        pwd="";
    }
    for(int i=begin;i<line.length();i++)pwd+=line[i];
    if(check(pwd))cout<<pwd;
    return 0;
}

```