GESP C++ 四级模拟试题 1

一、 选择题

1. （2023年6月）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

   【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. （2023年12月）某公司新出了一款无人驾驶的小汽车，通过声控智能驾驶系统，乘客只要告诉汽车目的地，车子就能自动选择一条优化路线，告诉乘客后驶达那里。请问下面哪项**不是**驾驶系统完成选路所必须的(　)。

   A. 麦克风

   B. 扬声器

   C. 油量表

   D. 传感器

   【答案】C

   【考纲知识点】基础知识

   【解析】本题属于考察计算机基础知识、乘客通过声音告诉汽车目的地，需要麦克风，扬声器，传感器，油量表不是必须的，所以本题正确答案为 C。

3. （2023年12月）下列C++语句执行以后结果是true的是(　)。

   A. 3&&false

   B. 5&&2

   C. 101&&000

   D. 4&true

   【答案】B

   【考纲知识点】逻辑运算符

   【解析】本题属于考察逻辑与运算符，表达式3&&false的结果为false，表达式5&&2的结果为true，表达式101&&000的结果为false，&是按位与运算符，所以表达式4&true结果为false，所以本题正确答案为B。

4. （2023年6月）下列关于 C++语言中指针的叙述，**不正确**的是(　)。

   A. 指针变量中存储的是内存地址。

B. 定义指针变量时必须指定其指向的类型。

C. 指针变量只能指向基本类型变量，不能指向指针变量。

D. 指针变量指向的内存地址不一定能够合法访问。

【答案】C

【考纲知识点】指针

【解析】本题属于考察指针的基本概念；指针变量不仅可以指向基本类型的变量 也可以指向其它的指针变量，所以本题正确答案为 C。

5. （2023年6月）下列关于 C++语言中数组的叙述，**不正确**的是(　)。

A. 一维数组在内存中一定是连续存放的。

B. 二维数组是一维数组的一维数组。

C. 二维数组中的每个一维数组在内存中都是连续存放的。

D. 二维数组在内存中可以不是连续存放的。

【答案】D

【考纲知识点】二维及多维数组

【解析】本题属于考察二维数组的基本概念；数组（包括多维数组）在内存中必须要连续存放，所以本题正确答案为 D。

6. （2023年9月）下列关于C++语言中函数的叙述，正确的是(　)。

A. 函数调用前必须定义。

B. 函数调用时必须提供足够的实际参数。

C. 函数定义前必须声明。

D. 函数声明只能写在函数调用前。

【答案】B

【考纲知识点】函数的概念及使用

【解析】本题属于考察计算机函数知识。函数调用时如果缺少实参将不能正确运行，所以本题正确答案为 B。

7. （2023年6月）下列关于 C++语言中变量的叙述，正确的是(　)。

A. 变量定义后可以一直使用。

B. 两个变量的变量名不能是相同的。

C. 两个变量的变量名可以相同，但它们的类型必须是不同的。

D. 两个变量的变量名可以相同，但它们的作用域必须是不同的。

【答案】D

【考纲知识点】全局、局部作用域

【解析】本题属于考察变量定义域的基本概念；在 C++中两个变量可以取相同的变量名，只要它们在不同的作用域下即可，所以本题正确答案为 D。

8. （2023年9月）如果n为int类型的变量，一个指针变量定义为int *p=&n; ,则下列说法正确的是(　)。

A. 指针变量p的值与变量n是相同。

B. 指针变量p的值与变量n的地址是相同的。

C. 指针变量p指向的值为 'n'。

D. 指针变量p指向的值与变量n的地址是相同的。

【答案】B

【考纲知识点】指针

【解析】本题属于考察C++指针知识。指针的值保存的是变量的地址，所以本题正确答案为B。

9. （2023年6月）一个二维数组定义为 int array[5][3];，则 array[1][2]和 array[2][1]在内存中的位置相差多少字节？（ ）

A. 2字节

B. 4字节

C. 8字节

D. 无法确定

【答案】C

【考纲知识点】二维及多维数组

【解析】本题属于考察内存地址的基本概念；array[1][2]和 array[2][1]中间差了array[2][0]，相当于差了 2 个 int，也就是 8 字节，所以本题正确答案为 C。

10. （2023年9月）如果 a 为 int 类型的变量，且 a 的值为6，则执行a = ~a; 之后，a 的值会是( )。

A. -6

B. 6

C. -7

D. 7

【答案】C

【考纲知识点】位运算

【解析】本题属于考察C++位运算知识。6按位取反运算，注意符号位也取反，呈现的是补码，转换过来就是-7。具体过程：

00000110（取反操作）

11111001（补码）

11111000（补码-1=反码）

10000111（负数的原码，注意此时取反符号位不变）

11. （2023年6月）一个数组定义为 int a[5] = {1, 2, 3, 4, 5};，一个指针定义为 int * p = &a[2];，则 执行 a[1] = *p;后，数组 a 中的值会变为( )。

A. {1, 3, 3, 4, 5}

B. {2, 2, 3, 4, 5}

C. {1, 2, 2, 4, 5}

D. {1, 2, 3, 4, 5}

【答案】A

【考纲知识点】指针

【解析】本题属于考察指针的基本概念；首先让指针 p 指向变量 a[2]的内存地址，然后让 a[1]=*p，也就是让 a[1]=a[2]，所以 a 数组变为 {1, 3, 3, 4, 5}。所以本题正确答案为 A。

12. （2023年9月）下列关于C++语言中异常处理的叙述，正确的是(　)。

A. 一个try子句可以有多个catch子句与之对应。

B. 如果try子句在执行时发生异常，就一定会进入某一个catch子句执行。

C. 如果try子句中没有可能发生异常的语句，会产生编译错误。

D. catch子句处理异常后，会重新执行与之对应的try子句。

【答案】A

【考纲知识点】异常处理

【解析】本题属于考察C++处理异常知识。B选项中，得到对应类型中的异常才能catch操作，C选项中，当try子句中没有可能发生异常的语句，不会产生编译错误，D选项中，catch子句处理异常后，再跳转到最后一个 catch 块后面继续执行。所以本题正确答案为A。

13. （2023年6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;
void xchg(__________){//在此处填入代码
    int t=*x;
    *x=*y;
    *y=t;
}
int main(){
    int a=10, b=20;
    xchg(&a,&b);
    cout<<a<<" "<<b<<endl;
    return 0;
}
```

A. int x, int y

B. int * x, int * y

C. int a, int b

D. int & a, int & b

【答案】B

【考纲知识点】函数、指针

【解析】本题属于考察指针的基本概念； 题目要求输出 20 10，也就是把 a 和 b进行交换，参数中传递了 a 和 b 的内存地址，需要使用相应类型的指针来存放，所以本题正确答案为 B。

14. （2023年12月）下列C++代码输入1, 2, 3, 4，执行后，将输出的是(　)。

```
string str="";
cin>>str;
int strlen=str.length();
for(int i=0; i<strlen; i++){
    if(str[i]<='9'&&str[i]>='0'){
        cout<<str[i];
    }else{
        cout<<"#";
    }
}
```

A. 1#4#

B. 1#3#

C. 1#2#3#4#

D. 1#2#3#4

【答案】D

【考纲知识点】循环和字符串的知识

【解析】本题属于考察C++循环结构和字符串的知识。循环结构的作用是遍历字符串中的每个字符，数字字符正常输出，如果不是数字字符则输出#，输入的逗号是非数字字符，所以3个逗号的位置输出#。所以本题正确答案为D。

15. （2023年6月）在下列代码的横线处填写(　)，完成对有 n 个 int 类型元素的数组 array **由小到大**排序。

```
void SelectionSort(int array[], int n){
    int i,j,min,temp;
    for(int i=0; i<n-1; i++){
        min=i;
        for(int j=i+1; j<n; j++)
            if(_____)//在此处填写代码
                min=j;
        temp=array[min];
        array[min]=array[i];
        array[i]=temp;
    }
}
```

A. array[min] > array[j]

B. array[min] > array[i]

C. min > array[j]

D. min > array[i]

【答案】A

【考纲知识点】排序算法

【解析】本题属于考察选择排序算法；选择排序每次会从待排序的数据元素中选出最小的一个元素，存放在序列的起始位置，也就是对于所有的 i+1<=j<n，找到最小的 array[j]，所以本题正确答案为 A。

二、 判断题

1. （2023年12月）C++的内置函数 sort() 支持数组的局部排序。例如 int a={10,9,8,7,6,5,4,3,2,1}，可以用 sort(a,a+5) ,排序成 {6,7,8,9,10,5,4,3,2,1} 。

【答案】正确

【考纲知识点】sort函数

【解析】本题是C++中的sort函数，sort(a,a+5)，代表排序从a[0]开始，一直到a[4]，是对10,9,8,7,6这几个元素进行升序排序，因此排序后的结果为{6,7,8,9,10,5,4,3,2,1}。所以本题正确。

2. （2023年6月）数列 1, 1, 2, 3, 5, 8 ... 是以意大利数学家列昂纳多·斐波那契命名的数列，从第 三个数开始，每个数是前面两项之和。如果计算该数列的第 n 项（其中 n>3） fib(n)，我们采用如下方法：① 令 fib(1)=fib(2)=1 ②用循环 for i=3 to n 分别计 算 f(i) ③输出 fib(n)。这体现了递推的编程思想。

【答案】正确

【考纲知识点】递推算法

【解析】本题属于考察递推相关概念，递推是按照一定的规律来计算序列中的每个项，本题中规律是从第三个数开始，每个数是前面两项之和，且我们按照从小到大的顺序依次计算数列中的每个项，这和递归的编程思想一致，所以本题正确。

3. （2023年9月）在C++语言中，每个变量都有其作用域。

【答案】正确

【考纲知识点】变量的作用域

【解析】本题是C++变量的知识，变量都有作用域。所以本题正确。

4. （2023年6月）在 C++语言中，函数的参数默认以引用传递方式进行传递。

【答案】错误

【考纲知识点】函数

【解析】本题属于考察函数相关概念，函数的参数默认以值传递方式进行传递，所以本题错误。

5. （2023年9月）在C++语言中，可以通过定义结构体，定义一个新的数据类型。

【答案】正确

【考纲知识点】结构体

【解析】本题是C++语言的知识，定义结构体可以认为定义一个新的数据类型。所以本题正确。

6. （2023年6月）如果希望记录 10 个最长为 99 字节的字符串，可以将字符串数组定义为 char s[100][10];。

【答案】错误

【考纲知识点】二维及多维数组

【解析】本题属于考察数组相关概念。最长为 99 个字节的字符串，应申请 100个 char 的数组；要定义 10 个最长为 99 字节的字符串，应该将字符串数组定义为 char s[10][100]，所以本题错误。

7. （2023年12月）小杨最近在准备考GESP，他用的Dev C++来练习和运行程序，所以Dev C++也是一个小型操作系统。

【答案】错误

【考纲知识点】计算机基础知识

【解析】本题属于考察计算机基础知识，DEVC++是一个编辑器，并不是操作系统。所以本题错误。

8. （2023年6月）字符常量'0'和'\0'是等价的。

【答案】错误

【考纲知识点】字符串

【解析】本题属于考察字符串相关概念，'0'是一个字符常量，它的 ASCII 码值为48；'\0'也是一个字符常量，它的 ASCII 码值为 0，通常用来表示字符串或字符数组的结束标志。可见它们不等价，所以本题错误。

9. （2023年12月）执行C++代码 cout<<(5||2)的结果为1。

【答案】正确

【考纲知识点】逻辑运算

【解析】本题属于考察C++中的逻辑运算，表达式5||2的结果为1，所以本题正确。

10. （2023年6月）由于文件重定向操作，程序员在使用 C++语言编写程序时无法确定通过 cout 输出的内容是否会被输出到屏幕上。

【答案】正确

【考纲知识点】文件操作

【解析】本题属于考察文件操作相关概念。使用文件重定向操作后，cout 输出的内容可能被写入文件而不是屏幕上。这是由程序用户决定的，编写程序的程序员无法确定，所以本题正确。

三、 编程题

1. 进制转换（2023年9月）

【问题描述】

进制数指的是逢 N 进一的计数制。例如，人们日常生活中大多使用十进制计数，而计算机底层则一般使用二进制。除此之外，八进制和十六进制在一些场合也是常用的计数制（十六进制中，一般使用字母 A 至 F 表示十至十五；本题中，十一进制到十五进制也是类似的）。

在本题中，我们将给出 N 个不同进制的数。你需要分别把它们转换成十进制数。

【提示】

对于任意一个 L 位 K 进制数，假设其最右边的数位为第 0 位，最左边的数位为第 L-1 位，我们只需要将其第 i 位的数码乘以权值 K^i，再将每位的结果相加，即可得到原 K

进制数对应的十进制数。下面是两个例子：

1. 八进制数 1362 对应的十进制数为 $1x8^3+3x8^2+6x8^1+2x8^0 = 754$

2. 十六进制数 3F0 对应的十进制为 $3x16^2+15x16^1+0x16^0 = 1008$

【输入描述】

输入的第一行为一个十进制表示的整数 N。接下来 N 行，每行一个整数 K，随后是一个空格，紧接着是一个 K 进制数，表示需要转换的数。保证所有 K 进制数均由数字和大写字母组成，且不以 0 开头。保证 K 进制数合法。

保证 N≤1000，保证 2≤K≤16。

保证所有 K 进制数的位数不超过 9。

【输出描述】

输出 N 行，每一个十进制数，表示对应 K 进制数的十进制数值。

【样例输入 1】

2

8 1362

16 3F0

【样例输出 1】

754

1008

【样例输入 2】

2

2 11011

10 123456789

【样例输出 2】

27

123456789

【题目大意】有n个k进制的整数，将它们分别转换成对应的十进制。

【考纲知识点】基本运算、输入输出语句、循环、进制转换的知识。

【解题思路】

1. 按题目要求定义好需要的变量，并实现输入；

2. 输入n行，每行2个整数，分别表示进制和要转换的数字；

3. 按求进制方法，例如：$(abc)_2=a*2^2+b*2^1+c*2^0$；

【参考代码】

```cpp
#include <bits/stdc++.h>
using namespace std;
int trans_digit(char c){
    if(c<='9') return(c-'0');
    return (c-'A'+10);
}
long long trans(int k, char str[]){
    int l=strlen(str);
    long long res=0, pw=1;
    for(int i=l-1; i>=0; i--){
        res+=pw*trans_digit(str[i]);
        pw*=k;
    }
    return res;
}
int main(){
    int n=0;
    cin>>n;
    for(int t=0;t<n;t++){
        int k=0;
        char str[10];
        cin>>k>>str;
        cout<<trans(k,str)<<endl;
    }
    return 0;
}
```

2. 图像压缩（2023年6月）

【问题描述】

图像是由很多的像素点组成的。如果用 0 表示黑，255 表示白，0 和 255 之间的值代表不同程度的灰色，则可以用一个字节表达一个像素（取值范围为十进制 0-255、十六进制 00-FF）。这样的像素组成的图像，称为 **256 级灰阶**的灰度图像。

0 **255**

现在希望将 256 级灰阶的灰度图像压缩为 16 级灰阶，即每个像素的取值范围为十进制 0-15、十六进制 0-F。**压缩规则为：**

① 统计出每种灰阶的数量，取数量**最多的前 16 种灰阶**（如某种灰阶的数量与另外一种灰阶的数量相同，则以灰阶值**从小到大**为序），分别编号 **0-F**（最多的编号为 0，以此类推）。

② 其他灰阶转换到**最近的** 16 种灰阶之一，将某个点灰阶数与 16 种灰阶中的一种相减，绝对值**最小**即为最近，如果绝对值相等，则编号较小的灰阶更近。

【输入描述】

输入第 1 行为一个正整数 N，表示接下来有 N 行数据组成一副 256 级灰阶的灰度图像。约定 **10≤N≤20**。

第 2 行开始的行，每行为长度相等且为**偶数**的**字符串**，每**两个字符**用十六进制表示一个像素。约定输入的灰度图像**至少**有 16 种灰阶。约定每行**最多** 20 个像素。

【输出描述】

第一行输出压缩选定的 16 种灰阶的十六进制编码，共计 32 个字符。

第二行开始的行，输出压缩后的图像，每个像素一位十六进制数表示压缩后的灰阶值。

【样例输入】

10
00FFCFAB00FFAC09071B5CCFAB76
00AFCBAB11FFAB09981D34CFAF56
01BFCEAB00FFAC0907F25FCFBA65
10FBCBAB11FFAB09981DF4CFCA67
00FFCBFB00FFAC0907A25CCFFC76
00FFCBAB1CFFCB09FC1AC4CFCF67
01FCCBAB00FFAC0F071A54CFBA65
10EFCBAB11FFAB09981B34CFCF67
01FFCBAB00FFAC0F071054CFAC76
1000CBAB11FFAB0A981B84CFCF66

【样例输出】

ABCFFF00CB09AC07101198011B6776FC
321032657CD10E
36409205ACC16D
B41032657FD16D
8F409205ACF14D
324F326570D1FE
3240C245FC411D
BF4032687CD16D
8F409205ACC11D
B240326878D16E
83409205ACE11D

【样例解释】

灰阶'AB'、'CF'和'FF'出现 14 次，'00'出现 10 次，'CB'出现 9 次，'09'出现 7 次，'AC'出现 6 次，'07'出现 5 次，'10'、'11'和'98'出现 4 次，'01'、'1B'、'67'、'76'和'FC'出现 3 次。

【题目大意】先根据输入将输入的十六进制两两成对转换为十进制 0~255 之间的数，再取出现次数较多的前 16 个数作为标准“灰阶”，将其他的数根据与标准“灰阶”作差的绝对值大小就近转换为对应“灰阶”，最后再将灰阶转换为 16 进制输出。

【考纲知识点】多层循环、模拟法、函数的定义与调用

【解题思路】

1. 模拟题，我们按照题目的要求进行模拟即可，首先将数据输入，并把 16 进制转换为 10 进制；
2. 接着我们找出出现次数最多的 16 种灰阶，并给它们编号为0-F。
3. 然后遍历所有的灰阶，找出距离它们各自最近的 16 种灰阶之一。
4. 先输出出现次数最多的 16 种灰阶，记得转换为 16 进制，再输出压缩后的图像，即所有的灰阶都输出距离它们各自最近的 16 种灰阶之一。

【参考代码】

```
#include <bits/stdc++.h>
using namespace std;
// 一位十六进制字符转换为数字
int trans(char a) {
    if (a<='9')
        return (a-'0');
    return (a-'A'+10);
}
// 一位十六进制数字转换为字符
char itrans(int n) {
    if (n>=10)
        return (char)(n-10+'A');
    return (char)(n+'0');
}
int image[25][25];
int cpimg[25][25];
int his[256];
int color[16];
// 计算整数 c 的压缩值
int compress(int c){
    int min=255,min_i=0;
    for(int h=0;h<16;h++){
        int k=abs(c-color[h]);
        if(k<min){
            min=k;
            min_i=h; //下标为转换值
        }
    }
    return min_i;
}
```

```
int main(){
    int n;
    cin>>n;
    char line[50]={0};
    for(int i=0;i<n;i++){
        cin>>line;
        for(int j=0;j<strlen(line);j+=2){
            int c = trans(line[j])*16+trans(line[j+1]);
            image[i][j/2]=c;
            his[c]++;// 统计灰阶值出现的次数
        }
    }
    // 在 his 数组中找出最多的前 16 种灰阶值保存在数组 color
中
    for(int i=0;i<16;i++){
        int max=0;
        for(int j=0;j<256;j++){
            if(his[max]<his[j]){
                max=j;
            }
        }
        color[i]=max;
        his[max]=-1;// 标记为-1，该灰阶值已获取
    }
    // 对 image 数组中的元素进行压缩处理
    int m=strlen(line)/2;
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            cpimg[i][j]=compress(image[i][j]);
        }
    }
    // 输出选取的 16 个灰阶
    for (int i=0;i<16;i++)
        cout<<itrans(color[i]/16)<<itrans(color[i]%16);
    cout<<endl;
    // 输出压缩后的图像
    for (int i=0;i<n;i++) {
        for (int j=0;j<m;j++)
            cout<<itrans(cpimg[i][j]);
        cout<<endl;
    }
    return 0;
}
```