

GESP C++ 四级模拟试题 2

一、 选择题

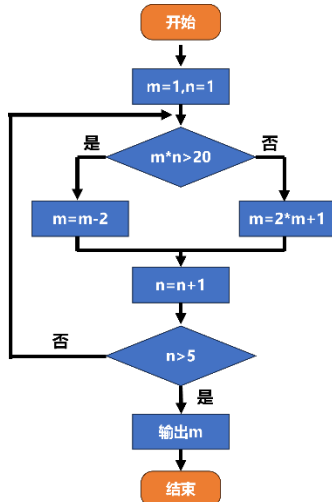
1. (2023年9月) 人们所使用的手机上安装的App通常指的是()。
- A. 一款操作系统
 - B. 一款应用软件
 - C. 一种通话设备
 - D. 以上都不对

【答案】B

【考纲知识点】计算机基础

【解析】本题属于考察计算机基础知识。APP是英语单词application的简写，代表应用软件的意思。所以本题正确答案为B。

2. (2023年9月) 下列流程图的输出结果是?()



- A. 9
- B. 7
- C. 5
- D. 11

【答案】A

【考纲知识点】流程图和多层循环语句

【解析】本题属于考察计算机循环结构嵌套选择结构知识。根据流程图， $n \leq 5$ 时会一直循环，每次循环先判断是否满足 $m*n > 20$ 的条件，然后对 m 进行不同的赋值。例如： $n=1, m=1, m*n$ 不大于20， m 会计算，赋值为3，依次类推，最终 m 的结果是9，所以本题正确答案为A。

3. (2023年12月) 下面C++代码执行后，输出的是()

```
int arr[10]={1};
string strArr="chen a dai";
cout<<strArr[arr[1]]<<endl;
```

- A. chen
- B. c
- C. chen a dai
- D. dai

【答案】B

【考纲知识点】数组、字符串

【解析】本题属于考察数组结合字符串的知识，定义数组arr时，给arr[0]赋初始值为1，其余元素为0，因此arr[1]的值为0，strArr[arr[1]]获取的是字符串中下标为0的字符，为c，所以本题正确答案为B。

4. （2023年6月）排序算法是稳定的（Stable Sorting），就是指排序算法可以保证，在待排序数据中有两个相等记录的关键字 R 和 S（R 出现在 S 之前），在排序后的列表中 R 也一定在 S 前。下面关于排序稳定性的描述，正确的是（ ）。
- A. 冒泡排序是不稳定的。
 - B. 插入排序是不稳定的。
 - C. 选择排序是不稳定的。
 - D. 以上都不正确。

【答案】C

【考纲知识点】排序算法

【解析】本题属于考察排序的基本概念；冒泡排序、插入排序一般是稳定的，而选择排序一般是不稳定的，所以本题正确答案为C。

5. （2023年9月）下列关于C++语言中指针的叙述，**不正确**的是（ ）。
- A. 可以定义指向int类型的指针。
 - B. 可以定义指向自定义结构体类型的指针。
 - C. 自定义结构体类型可以包含指针类型的元素。
 - D. 不能定义指向void类型的指针，那没有意义。

【答案】D

【考纲知识点】指针的概念及使用

【解析】本题属于考察计算机指针知识。可以指向void类型，所以本题正确答案为D。

6. （2023年9月）下列关于 C++语言中数组的叙述，**不正确**的是（ ）。
- A. 一维数组可以用来表示数列。
 - B. 二维数组可以用来表示矩阵。
 - C. 三维数组可以用来表示空间中物体的形状。
 - D. 世界是三维的，所以定义四维数组没有意义。

【答案】D

【考纲知识点】数组

【解析】本题属于考察计算机数组知识。世界可以用多维描述，定义四维也有意义，所以本题正确答案为 D。

7. (2023年6月) 下列关于C++语言中函数的叙述, 正确的是()。

- A. 函数必须有名字。
- B. 函数必须有参数。
- C. 函数必须有返回值。
- D. 函数定义必须写在函数调用前。

【答案】A

【考纲知识点】函数

【解析】本题属于考察函数的基本概念; 函数可以没有参数和返回值, 同时函数定义可以在调用之后, 只需要在调用前加函数声明, 但是函数必须要有名字, 所以本题正确答案为 A。

8. (2023年12月) 在C++中, 执行下面代码后, 输出的是()。

```
int point(int *p){  
    return *p **p;  
}  
  
int main(){  
    int a=20;  
    int *p=&a;  
    *p=point(p);  
    cout<<*p<<endl;  
    return 0;  
}
```

- A. 400
- B. 200
- C. 20
- D. 100

【答案】A

【考纲知识点】指针

【解析】本题属于考察指针的应用; *p保存的是a的地址, 指向变量a, 调用point函数, 传入p, 代表传入a的地址, 形参*p保存变量a的地址, 指向变量a, 因此*p的值为20, 所以*p与*p相乘的结果为400, point函数的返回值为400。所以本题正确答案为 A。

9. (2023年6月) 一个变量定义为 int *p = nullptr;, 则下列说法正确的是()。

- A. 该指针变量的类型为 int。
- B. 该指针变量指向的类型为 int。
- C. 该指针变量指向的内存地址是随机的。
- D. 访问该指针变量指向的内存会出现编译错误。

【答案】B

【考纲知识点】指针

【解析】本题属于考察指针的基本概念。指针变量的类型为 `int *`，A 选项错误；`nullptr` 指向的是固定的内存地址 0，C 选项错误；`nullptr` 指向的内存位置通常不存放有效数据，因此常用来表示未指向有效数据，访问它可能出现运行时错误，但不会出现编译错误，D 选项错误。本题正确答案为 B。

10. (2023年9月) 一个三维数组定义为 `long long array[6][6][6]`；，则 `array[1][2][3]` 和 `array[3][2][1]` 在内存中的位置相差多少字节？（ ）
- A. 70字节
 - B. 198字节
 - C. 560字节
 - D. 无法确定

【答案】C

【考纲知识点】三维数组

【解析】本题属于考察C++三维数组知识。数组是`long long`类型，每间隔一下数字就间隔8个字节，假设三维数组用长宽高来描述，`array[0][0][0]`到`array[0][5][5]`有36个元素，`array[1][2][3]`包含 $36+12+4=52$ 个元素；到长度3宽度2高度1，包括 $36*3+12+2=122$ 个元素，隔离70个元素，因此长度是560，所以本题正确答案为C。

11. (2023年9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 `*p = a[1]`；后，数组a中的值会变为（ ）。
- A. {1, 2, 2, 4, 5}
 - B. {1, 3, 3, 4, 5}
 - C. {1, 2, 3, 3, 5}
 - D. {1, 2, 4, 4, 5}

【答案】A

【考纲知识点】指针和数组

【解析】本题属于考察C++指针和数组知识。P指针指向的是`a[2]`，将`a[1]`的值赋值给p指向的变量`a[2]`，数组中`a[1]`将等于`a[2]`。所以本题正确答案为A。

12. (2023年6月) 以下哪个函数声明在调用时可以传递二维数组的名字作为参数？（ ）。
- A. `void BubbleSort(int a[][4]);`
 - B. `void BubbleSort(int a[3][]);`
 - C. `void BubbleSort(int a[][]);`
 - D. `void BubbleSort(int ** a);`

【答案】A

【考纲知识点】函数、指针、二维及多维数组

【解析】本题属于考察函数参数的基本概念。当把数组作为函数的一个参数时，实际上只传递了数组的首指针。于是，传递多维数组时，只有形式参数的第一维的长度可以省略，形式参数的其他维的长度都不能省略。所以本题正确答案为 A。

13. (2023年12月) 以下C++代码用于实现每个整数对应的因数，如输入12，则输出

1 2 3 4 6 12; 如输入18, 则输出2 3 6 9 18。横线处应填入代码是()。

```
int n;  
cin>>n;  
for(int i=1; i<=n; i++){  
    _____{//此处填写代码  
        cout<<i<<" ";  
    }  
}
```

- A. if(n%i==0)
- B. if(n/i==0)
- C. if(n%i!=0)
- D. if(n/i!=0)

【答案】A

【考纲知识点】选择和循环结构

【解析】本题属于C++中的选择和循环结构。程序的目的是找出1~n之间, n的所有因数, 因此需要判断n对每个数字取余是否为0。所以本题正确答案为A。

14. (2023年6月) 执行以下 C++语言程序后, 输出结果是()。

```
#include <iostream>  
using namespace std;  
int main(){  
    int array[3][3];  
    for(int i=0; i<3; i++)  
        for(int j=0; j<3; j++)  
            array[i][j]=i*10+j;  
    int sum;  
    for(int i=0; i<3; i++)  
        sum+=array[i][i];  
    cout<<sum<<endl;  
    return 0;  
}
```

- A. 3
- B. 30
- C. 33
- D. 无法确定

【答案】D

【考纲知识点】全局、局部作用域

【解析】本题属于考察变量初始化相关概念; 因为 sum 是在函数内部定义的, 所以 sum 的初始值并不一定是 0, 也就无法确定最终的输出了, 所以本题正确答案为 D。

15. (2023年9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 BubbleSort(int array[], int n){  
    for (int i=n;i>2;i--)  
        for ( ) // 在此处填入代码  
            if(array[j]>array[j+1]){  
                int t=array[j];  
                array[j]=array[j+1];  
                array[j+1]=t;  
            }  
}
```

- A. int j =1; j < n; j++
- B. int j =0; j < n; j++
- C. int j =0; j<i-1; j++
- D. int j =0; j<i; j++

【答案】C

【考纲知识点】函数和数组的知识

【解析】本题属于考察C++函数和数组的知识。题目已知，要求从小到大的冒泡排序，如果前一个比后一个大，就交换位置，外层循环从n开始，判断的区间逐渐减小，判断循环范围C正确，所以本题正确答案为C。

二、 判断题

1. (2023年12月) C++内置函数 sort() 可以对整数、浮点数、字符数组进行从大到小，从小到大，局部排序。。

【答案】正确

【考纲知识点】sort函数

【解析】本题属于考察sort函数，sort函数可以对整数、浮点数、字符数组进行排序，可以升序、降序或者是局部排序。所以本题正确。

2. (2023年9月) 对N个元素的数组执行插入排序算法，通常的时间复杂度是 $O(N^2)$ 。

【答案】正确

【考纲知识点】排序算法

【解析】本题是C++排序算法的知识，插入排序常用时间复杂度是 $O(N^2)$ ，所以本题正确。

3. (2023年6月) 在 C++语言中，可以定义四维数组，但在解决实际问题时不可能用到，因为世界是三维的。

【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。实际问题中是有可能使用到四维甚至更多维数组的，所以本题错误。

4. (2023年6月) 在 C++ 语言中, 一个函数没有被调用时, 它的参数不占用内存。
- 【答案】正确
- 【考纲知识点】函数
- 【解析】本题属于考察函数相关概念。函数的参数只有在函数被调用时才会在调用栈上分配对应内存, 并在函数返回时回收, 这也形成了函数参数的生命周期。因此, 函数的参数在没有被调用时不会占用内存, 所以本题正确。
5. (2023年12月) 在C++中, 两个字符串相加的运算符为+相当于字符串的合并运算。下面C++代码执行后, 将输出 chenadai。
- ```
string a="chen";
string b="a";
string c="dai";
string name=a+b+c;
cout<<name<<endl;
```
- 【答案】正确
- 【考纲知识点】字符串运算
- 【解析】本题是C++字符串运算, a、b、c三个字符串相加相当于拼接三个字符串, 因此输出结果是正确的。所以本题正确。
6. (2023年9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[10][100];。
- 【答案】正确
- 【考纲知识点】数组
- 【解析】本题是C++字符数组的知识, 定义长度大于等于要求。所以本题正确。
7. (2023年6月) 在 C++ 语言中, 如果一个函数可能抛出异常, 那么一定要在 try 子句里调用这个函数。
- 【答案】错误
- 【考纲知识点】异常处理
- 【解析】本题属于考察异常处理相关概念, 即使一个函数可能抛出异常, 也不一定要在 try 子句里调用这个函数。可正常调用, 异常会向调用更上层抛出, 如上层调用在 try 子句中, 则可以在上层捕获处理。所以本题错误。
8. (2023年12月) 任何一个while循环都可以转化为等价的for循环。
- 【答案】正确
- 【考纲知识点】循环结构
- 【解析】本题考查循环结构。while循环能够实现的代码, 使用for循环也能够实现。所以本题正确。
9. (2023年6月) >=和>>=都是 C++ 语言的运算符。
- 【答案】正确
- 【考纲知识点】基本运算、位运算
- 【解析】本题属于考察运算符相关概念。>=是关系运算符大于等于; >>=是复合

位右移赋值运算符,  $a \gg= b$ ;等价于  $a = a \gg b$ ;。所以本题正确。

10. (2023年9月) 通过使用文件重定向操作, 可以将程序中输出到cout的内容输出到文件中, 这是常用的记录程序运行日志的方法之一。

【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知识。所以本题正确。

### 三、编程题

1. 幸运数 (2023年6月)

【问题描述】

小明发明了一种“幸运数”。一个正整数, 其偶数位不变(个位为第1位, 十位为第2位, 以此类推), 奇数位做如下变换: 将数字乘以7, 如果不大于9则作为变换结果, 否则把结果的各位数相加, 如果结果不大于9则作为变换结果, 否则(结果仍大于9)继续把各位数相加, 直到结果不大于9, 作为变换结果。变换结束后, 把变换结果的各位数相加, 如果得到的和是8的倍数, 则称一开始的正整数为幸运数。

例如, 16347: 第1位为7, 乘以7结果为49, 大于9, 各位数相加为13, 仍大于9, 继续各位数相加, 最后结果为4; 第3位为3, 变换结果为3; 第5位为1, 变换结果为7。最后变化结果为76344, 对于结果76344其各位数之和为24, 是8的倍数。因此16347是幸运数。

【输入描述】

输入第一行为正整数N, 表示有N个待判断的正整数。约定  $1 \leq N \leq 20$ 。

从第2行开始的N行, 每行一个正整数, 为待判断的正整数。约定这些正整数小于  $10^{12}$ 。

【输出描述】

输出N行, 对应N个正整数是否为幸运数, 如是则输出'T', 否则输出'F'。

提示: 不需要等到所有输入结束在依次输出, 可以输入一个数就判断一个数并输出, 再输入下一个数。

【样例输入】

2  
16347  
76344

【样例输出】

T  
F

【题目大意】

对正整数的奇数位进行乘以7, 进行若干次数位求和小于等于9之后, 再与偶数位求和, 判断结果是否能被8整除。

【考纲知识点】模拟法、函数的定义和调用



### 【解题思路】

1. 首先读入所有待判断的正整数，并依次进行判断，注意使用 long long 类型；
2. 接着通过将个数不停整除 10 来得到它的每一位；
3. 对其中的奇数位按照题目的要求进行变换，偶数位则不变。奇数位的变换可以按题目文字循环处理，也可以找到等价的更简洁的表达式；
4. 累加每一位并判断总和是否能被 8 整除。

注意在解题过程中合理设计和调用函数，可以使得思路更清晰。

### 【参考代码】

```
#include <bits/stdc++.h>
using namespace std;
int trans(int t){
 if(t==0) return 0;
 return (t*7-1)%9+1;
}
bool judge(long long x){
 int sum=0;
 for(int d=1; x>0; d++, x/=10){
 int t=(int)(x%10);
 if(d%2==0) sum+=t;
 else sum+=trans(t);
 }
 return (sum%8==0);
}
int main() {
 int N=0;
 cin>>N;
 for(int n=0; n<N; n++){
 long long x=0;
 cin>>x;
 if(judge(x)) cout<<"T"<<endl;
 else cout<<"F"<<endl;
 }
 return 0;
}
```

## 2. 变长编码 (2023年9月)

### 【问题描述】

小明刚刚学习了三种整数编码方式：原码、反码、补码，并了解到计算机存储整数通常使用补码。但他总是觉得，生活中很少用到  $2^{31}-1$  这么大的数，生活中常用的 0~100 这种数也同样需要 4 个字节的补码表示，太浪费了些。热爱学习的小明通过搜索，发现了一种正整数的变长编码方式。这种编码方式的规则如下：

1. 对于给定的正整数，首先将其表达为二进制形式。例如， $(0)_{(10)} = (0)_{(2)}$ ， $(926)_{(10)} = (1110011110)_{(2)}$ 。
  2. 将二进制数从低位到高位切分成每组 7bit，不足 7bit 的在高位用 0 填补。例如， $(0)_{(2)}$  变为 0000000 的一组， $(1110011110)_{(2)}$  变为 0011110 和 0000111 的两组。
  3. 由代表低位的组开始，为其加入最高位。如果这组是最后一组，则在最高位填上 0，否则在最高位填上 1。于是，0 的变长编码为 00000000 一个字节，926 的变长编码为 10011110 和 00000111 两个字节。
- 这种编码方式可以用更少的字节表达比较小的数，也可以用很多的字节表达非常大的数。例如，987654321012345678 的二进制为  $(0001101\ 1011010\ 0110110\ 1001011\ 1110100\ 0100110\ 1001000\ 0010110\ 1001110)_{(2)}$ ，于是它的变长编码为（十六进制表示）CE 96 C8 A6 F4 CB B6 DA 0D，共 9 个字节。

你能通过编写程序，找到一个正整数的变长编码吗？

**【输入描述】**

输入第一行，包含一个正整数 N。约定  $0 \leq N \leq 10^{18}$ 。

**【输出描述】**

输出一行，输出 N 对应的变长编码的每个字节，每个字节均以 2 位十六进制表示（其中 A-F 使用大写字母表示），两个字节间以空格分隔。

**【样例输入 1】**

0

**【样例输出 1】**

00

**【样例输入 2】**

926

**【样例输出 2】**

9E 07

**【样例输入 3】**

987654321012345678

**【样例输出 3】**

CE 96 C8 A6 F4 CB B6 DA 0D

**【题目大意】** 给一个正整数，根据题目要求找它的变长编码，变长编码用 16 进制表示。

**【考纲知识点】** 基本运算、输入输出语句、一维数组、位运算的知识

**【解题思路】**

1. 按题目要求定义好需要的变量，并实现输入；
2. 根据题意，将 n 对应的二进制，每 7 位 1 组，保存起来。
3. 除了包含最高位那组，其他组最前面都增加 1，例如中间一组是 0001111，最前面加 1 变成 10001111；
4. 注意输出是先输出低位的数组，输出内容用 16 进制表示即可。

**【参考代码】**

```

#include <bits/stdc++.h>
using namespace std;
void output_digit(int d) {
 if (d >= 10)
 cout << (char)('A' + d - 10);
 else
 cout << (char)('0' + d);
}
void output_code(int s) {
 output_digit(s >> 4);
 output_digit(s & 0x0f);
}
int main() {
 long long n = 0;
 cin >> n;
 int split[10];
 int l = 0;
 while (n > 0) {
 split[l] = (int)(n & 0x7f);
 n >>= 7;
 l++;
 }
 for (int i = 0; i < l - 1; i++) split[i] |= 0x80;
 output_code(split[0]);
 for (int i = 1; i < l; i++) {
 cout << " ";
 output_code(split[i]);
 }
 return 0;
}

```