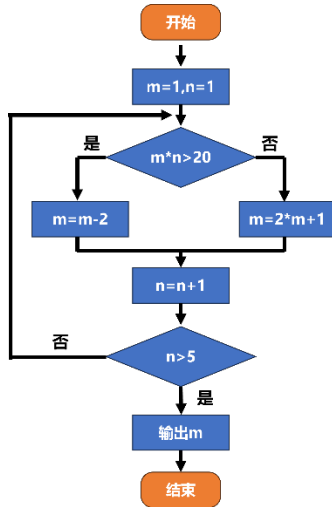


GESP C++ 四级模拟试题 2

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
2. (2023年9月) 下列流程图的输出结果是?()



- A. 9
B. 7
C. 5
D. 11
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
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。
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。
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。
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。
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
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。
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

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}

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);`

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)`

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

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++

## 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。。
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 $O(N^2)$ 。
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。
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。
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;

```

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];`。

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。
8. (2023年12月) 任何一个 while 循环都可以转化为等价的 for 循环。
9. (2023年6月) `>=` 和 `>>=` 都是 C++ 语言的运算符。
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。

### 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
```

#### 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