

每日一题C++方向day23_12月1日

一. 单选

1.

在双向循环链表中，在p指针所指的节点后插入一个指针q所指向的新节点，修改指针的操作是_____。

- ☐ A p->next=q;q->prior=p;p->next->prior=q;q->next=q;
- ☐ B p->next=q;p->next->prior=q;q->prior=p;q->next=p->next;
- ☐ C q->prior=p;q->next=p->next;p->next->prior=q;p->next=q;
- ☐ D q->next=p->next;q->prior=;p->next=q;p->next=q;

正确答案：C

2.

以下程序的运行结果是：

```
int main(void){
    int m=5;
    if(m++>5)cout<<m<<endl;
    else cout<<--m;
}
```

- ☐ A 4
- ☐ B 5
- ☐ C 7
- ☐ D 6

正确答案：B

3.

```
struct Date
{
    char a;
    int b;
    int64_t c;
    char d;
};
Date data[2][10];
```

在64位系统上，如果Data的地址是x，那么data[1][5].c的地址是（ ）

- ☐ A X+195
- ☐ B X+365
- ☐ C X+368
- ☐ D X+215

正确答案：C

4.

采用递归方式对顺序表进行快速排序，下列关于递归次数的叙述中，正确的是（ ）

- ☐ A 递归次数与初始数据的排列次序无关
- ☐ B 每次划分后，先处理较长的分区可以减少递归次数
- ☐ C 每次划分后，先处理较短的分区可以减少递归次数
- ☐ D 递归次数与每次划分后得到的分区处理顺序无关

正确答案：D

5. 一棵完全二叉树第六层有9个叶结点（根为第一层），则结点个数最多有（ ）

- ☐ A 112
- ☐ B 111
- ☐ C 107
- ☐ D 109

正确答案：D

6. 两个人两个小时能组装两辆自行车,要在6小时内组装12辆自行车,需要多少人?

- ☐ A 2
- ☐ B 3
- ☐ C 4
- ☐ D 5

正确答案：C

7. 已知关键字序列5,8,12,19,28,20,15,22是最小堆，插入关键字3，调整后得到的最小堆是()

- ☐ A 3,8,12,5,20,15,22,28,19
- ☐ B 3,5,12,19,20,15,22,8,28
- ☐ C 3,12,5,8,28,20,15,22,19
- ☐ D 3,5,12,8,28,20,15,22,19

正确答案：D

8. 已知一个线性表 (38 , 25 , 74 , 63 , 52 , 48) , 假定采用散列函数 $h(\text{key}) = \text{key} \% 7$ 计算散列地址, 并散列存储在散列表A【0...6】中, 若采用线性探测方法解决冲突, 则在该散列表上进行等概率成功查找的平均查找长度为

- A 1.5
- B 1.7
- C 2.0
- D 2.3

正确答案：C

9.
以30为基准,设一组初始记录关键字序列为 (30,15,40,28,50,10,70), 则第一趟快速排序结果为 ()

- A 10 , 28 , 15 , 30 , 50 , 40 , 70
- B 10 , 15 , 28 , 30,50 , 40 , 70
- C 10 , 28 , 15 , 30 , 40 , 50 , 70
- D 10 , 15 , 28 , 30 , 40 , 50 , 70

正确答案：B

10. 一棵二叉树的先序遍历为EFHIGJK, 中序遍历为HFIEJKG, 则后序遍历为 ()

- A HIFJKGE
- B FHIJKGE
- C HIFGJKE
- D HIFKJGE

正确答案：D

二. 编程

1. 标题：微信红包 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

春节期间小明使用微信收到很多个红包，非常开心。在查看领取红包记录时发现，某个红包金额出现的次数超过了红包总数的一半。请帮小明找到该红包金额。写出具体的算法思路和代码实现，要求算法尽可能高效。

给定一个红包的金额数组gifts及它的大小n，请返回所求红包的金额。

若没有金额超过总数的一半，返回0。

测试样例：

[1,2,3,2,2],5

返回：2

输入描述：

输出描述：

示例1:

输入

输出

正确答案：

2. 标题：计算字符串的距离 | 时间限制：1秒 | 内存限制：32768K

Levenshtein 距离，又称编辑距离，指的是两个字符串之间，由一个转换成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。编辑距离的算法是首先由俄国科学家Levenshtein提出的，故又叫Levenshtein Distance。

Ex：

字符串A:abcdefg

字符串B: abcdef

通过增加或是删掉字符" g"的方式达到目的。这两种方案都需要一次操作。把这个操作所需要的次数定义为两个字符串的距离。

要求：

给定任意两个字符串，写出一个算法计算它们的编辑距离。

请实现如下接口

/*

功能：计算两个字符串的距离

*

输入：

字符串A和字符串B

*

输出：无

```
*
```

返回：如果成功计算出字符串的距离，否则返回-1

```
*/
```

```
public
```

```
static
```

```
int
```

```
style="font-family: 'courier new';font-size: 13.0px;"> calStringDistance ( style="background:  
silver;">String charA, String charB)
```

```
{
```

```
return
```

```
0;
```

```
}
```

输入描述：
输入两个字符串

输出描述：
得到计算结果

示例1:

输入

abcdefg

abcdef

输出

1

正确答案：