验收卷(五)　综合练习(A)

(考试时间40分钟　满分50分)

一、选择题(本题共12小题，每小题2分，共24分)

1.下列关于数据结构与算法效率的描述，不正确的是(C)

A.队列和栈都是一种线性表，但两者有不相同的特性

B.采用相同公式求解n！，使用迭代算法比递归算法的算法效率高

C.使用数组结构在进行数据插入和删除操作时，一定会引起数据移动

D.某单向链表(节点数>2)设有头尾指针，在删除该链表尾节点时需要遍历多个节点

解析 本题考查不同数据结构的特性和算法描述。A选项队列和栈都是一种线性表，队列的特性是先进先出，栈的特性是先进后出。B选项递归求 n！要递推和回归两个阶段(分解问题，逐级返回)，迭代求 n！循环代码中参与运算的变量作为下一次循环计算的初始值。当 n 越大，迭代算法的效率明显高于递归算法。C选项使用数组这种数组结构在数组尾部进行数据插入和删除操作时，不会引起数据移动；D选项链表的操作需要从头指针开始，遍历到需要操作的节点，因为某单向链表(节点数>2)，所以在对该链表进行删除尾节点的操作时需要遍历从头指针开始到尾指针等多个节点。

2.约定：T操作是指在队列中1个元素出队后再入队，E操作是指将1个元素入队，P操作是指队列中1个元素出队，队首指针head和队尾指针tail初始值均为0。则经过EETPETEP系列操作后，队首指针head和队尾指针tail的值分别为(**D**)

A.3　4　　B.3　5　C.4　5　　D.4　6

解析　本题考查队列的基本操作。T操作既有入队，又有出队，因此共有6次入队，4次出队，每次出队和入队，指针分别加1。

3.如果一个栈初始时为空，且当前栈中的元素从栈底到栈顶依次为“d，e，f，g”，已知元素 p 已经出栈，则入栈顺序不可能是(D)

A.d，e，f，g，p　　B.p，d，e，f，g

C.d，e，p，f，g　　D.d，e，g，p，f

解析　本题主要考查的是栈的入栈和出栈操作。已知当前栈中的元素从栈底到栈顶依次为“d，e，f，g”，说明入栈顺序为“d、e、f、g”，而元素f的进栈顺序则没有要求，因为元素d进栈可直接出栈，因此，ABC选项都有可能，D选项不可能是入栈顺序，因为元素f应在元素g之前入栈。

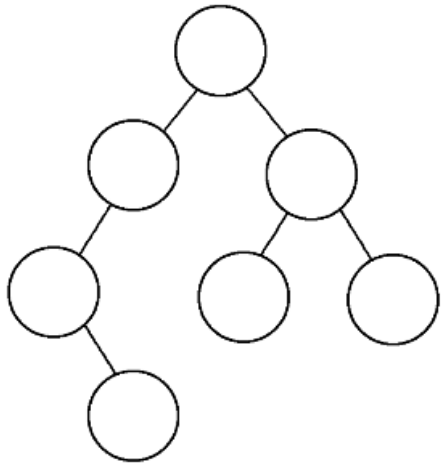
4.已知一棵完全二叉树的节点总数为 12，则有关该二叉树的说法正确的是(D)

A.树的度为12　　B.树的层数为3

C.叶子节点数为5　　D.最后一层有5个节点

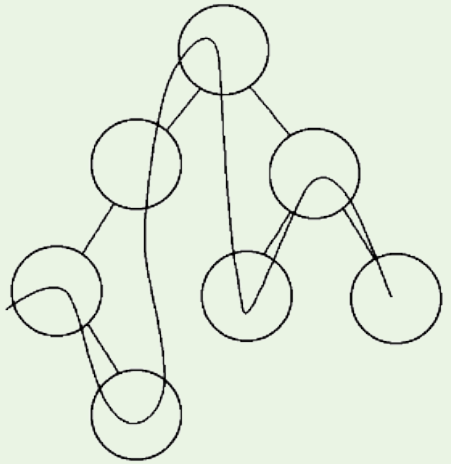
解析　本题考查二叉树的基本性质。符合完全二叉树的两个条件为：①只有最下二层中的节点度数小于2；②最下一层的叶子节点都依次排列在该层最左边位置。A选项度指树的最大节点数。B选项若为满二叉树，第3层及前面所有节点数和为7，因此至少4层。C选项最后一层上有5个节点，即有叶子节点数为5，但第3层上还有一个叶子节点。

5.某二叉树的树形结构如图所示，其中序遍历结果为FDGBAEC。若补全为完全二叉树后，按从上到下、自左往右的顺序用一维数组a存储，其中根节点存储于元素a[0]中，则元素a[6]的值为(**D**)



A.D　　B.F　　C.G　　D.C

解析　元素a[6]位于二叉树从上至下，从左到右第7个位置，即第3层最后一个。根据中序遍历画出图示如图所示。



6.有如下Python 程序段：

```
import random
q = [0]*5
head，tail = 0，0
for i in range(5)：
    if random.randint(0，i)%2 = = 0：
      q[tail] = random.randint(1，9) #随机生成1到9之间的整数
    elif head<tail and q[tail - 1]<q[head]：
        q[tail] = q[head]
       head + = 1
  tail + = 1
print(q)
```

执行该程序段后，列表 q 的值不可能是(C)

A.[1，0，1，0，9]　　B.[5，4，3，2，1]

C.[5，8，3，0，0]　　D.[5，5，6，0，6]

解析　循环5次，每次循环有3种可能，一是当i为偶数时，入队一个[1，9] 之间的整数，二是当i为奇数时，且q[tail－1]<q[head]，将队首出队后再入队，三是当i为偶数时，队不空，且q[tail－1]等于q[head]，入队一个0。A选项先入队1，由于q[tail－1]等于q[head]，入队0，新入队一个1、0、9。B选项5次均入队。C选项5入队后，当i为1时，q[tail－1]等于q[head]，只可能入队一个0。D选项5入队，5出队后入队，6入队，由于队中只有一个数，且i为奇数，因此0入队。接着6入队。

7.有如下 Python程序段：

```
a = [21，5，10，9，18，10，5，18，12，11]
n = len(a)
st = [0]*n; top = -1
for i in range(n)：
    if top = = -1：
        top + = 1
        st[top] = a[i]
    else：
        if a[i]%2 = = 0：
            while top> - 1 and a[i]>st[top]：
                top - = 1
```

```
        top + = 1
        st[top] = a[i]
while top> - 1 :
        print(st[top] , end = " ")
        top - = 1
```

执行该程序段后，输出结果为(**A**)

A.12 18 18 21　　B.18 18 12

C.21 18 18 12　　D.11 12 18 18 21

解析　本题考查栈的入栈和出栈。把握出入栈的条件，当栈为空(top＝＝－1)时入栈；当a[i]是偶数时，把栈顶中比该数小的数出栈，自身入栈。21入栈，10入栈，18先让10出栈，再入栈，18入栈，12入栈。

8.定义如下函数：

```
def chg(k)：
    if k = = - 1：
        return ""
    else：
      c = chr(ord("a") + k)
        if k%2 = = 1：
            return c + chg(k - 1)
    else：
        return chg(k - 1) + c
```

执行语句m = chg(4)后，m的值为(**B**)

A."ecabd"　　B."dbace"　　C."abcde"　　D."edcba"

解析　依次调用函数时，k的值为4，3，2，1，0，因此对应的字母c依次为edcba。chg(4)＝chg(3)＋"e"；chg(3)＝"d"＋chg(2)；chg(2)＝chg(1)＋"c"；chg(1)＝"b"＋chg(0)；chg(0)＝chg(－1)＋"a"，依次进行回归，chg(1)＝"ba"，chg(2)＝"bac"，chg(3)＝"dbac"，chg(4)＝"dbace"。

9.下面 Python 程序运行后，输出结果不可能的是(C)

```
from random import randint
a = [3，4，5，6，7，8]
def f(x)：
    if x<2：
        return a[x] + f(2*x + randint(1，3))
    else：
        return a[x]
```

A.8　　B.9　　C.10　　D.13

解析　若第1次产生的随机数为1，则返回a[0]＋f(2*0＋1)，即3＋f(1)。若x的值为1，函数返回a[1]＋f(2*1＋randint(1，3))，产生随机数分别为1，2，3时，返回值依次为4＋6＝10，4＋7＝11，4＋8＝12，则3＋f(1)的值依次为13，14，15。若第1次产生的随机数为2，则返回a[0]＋f(2*0＋2)，其值为3＋5＝8。若第1次产生的随机数为3，则返回a[0]＋f(2*0＋3)，其值为3＋6＝9。

10.列表a中存储了8个元素，即a[0]，a[1]，…，a[7]，有如下Python程序段：

```
n = 8
for i in range(n - 1)：
    for j in range(n - 1，i， - 1)：
        if a[j]<a[j - 1] and j%4！ = 0:
            a[j - 1]，a[j] = a[j]，a[j - 1]
```

该程序段实现的是(**D**)

A.a[0]到a[7]升序排序

B.a[4]到a[7]升序排序

C.a[0]到a[7]的数据对4取余之后升序排序

D.a[0]到a[3]、a[4]到a[7]分别升序排序

解析　a[7]和a[6]、a[6]和a[5]、a[5]和a[4]依次比较，实现a[4]到a[7]升序，j为4时，并没有和a[3]比较和交换，但a[3]和a[2]、a[2]和a[1]、a[1]和a[0]依次比较和交换，形成有序序列。

11.如下 Python 程序段：

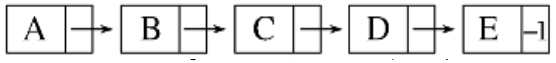
```
import random
a = [1，3，5，7，9，11，13，15]
key = random.randint(1，8)*2
i，j = 0，len(a) - 1
s = 0
while i<=j：
    m = (i + j + 1) // 2
    if a[m] == key：
```

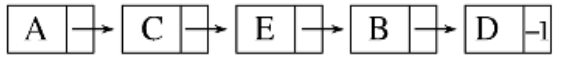
```
        break
    if a[m]>key：
        j = m - 1；s - = 1
    else：
        i = m + 1；s + = 1
print(s)
```

上述程序执行完以后，s 的值可能有(A)

A.4 种　　B. 5种　　C. 7种　　D. 8 种

解析　本题考查对分查找。列表 a 中的元素为1－15 中的奇数，查找的 key 为 2－16 中的偶数，所以查找的结果不存在 a[m]＝＝key 也就是找到的情况，break 不会被执行。查找的过程如下：s 的值可能为－2，－1，1，3，共4种。

12.将链表中的奇数节点和偶数节点分别排在一起，奇数节点和偶数节点的相对顺序不变。如原始链表为 



：

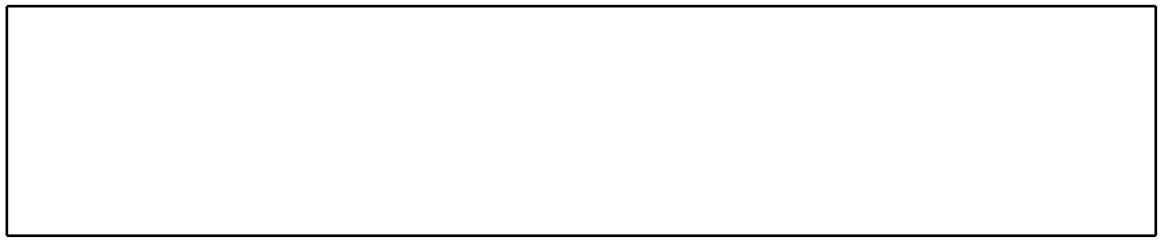
#读入链表，存储在列表a中，head存储链表的头节点

odd = head

even = a[odd][1]

tmp = even

while a[odd][1] ! = - 1 and a[even][1] ! = - 1 :



a[odd][1] = tmp

上述程序段中方框处可选的语句为

①odd = a[odd][1]

②even = a[even][1]

③a[odd][1] = a[even][1]

④a[even][1] = a[odd][1]

则方框处语句依次为(**D**)

A.①③②④　　B.①②③④　　C.③②④①　　D.③①④②

解析　本题考查链表的插入。先分别获取奇数节点连接而成的链表和偶数节点连接而成的链表；再连接两个链表得到新链表。

二、非选择题(本题共3小题，共26分)

13.(6分)疫情防控期间，某工厂为了将流水线上已生产的口罩及时装箱，并尽量分配给更多的疫情地区，需要设计一个程序实现自动化装箱。装箱要求为：流水线上生产的每包口罩数量有可能不同，装入箱子的口罩必须为流水线上连续的若干包，每箱内的口罩数量必须相同，在已知每包口罩数量和口罩总包数的前提下，装尽可能多的箱子。

例1：某流水线上有8包口罩，每包口罩的数量如下表：