

高三上信息技术 链表提高练习

【基础代码回顾】

链表的遍历 1: a=[[1, 1], [2, 2], [3, 3], [4, -1]] head=0; p=head while _____: print(a[p][0], end="") _____	链表的遍历 2: a=[[1, 1], [2, 2], [3, 3], [4, -1]] head=0 ; p=head while _____: print(a[p][0], end="->") _____ print(_____)
---	--

3. 在链表 a 中插入 b 节点形成升序序列:

```
a=[[12, 3], [18, -1], [10, 0], [14, 1]]
head=2 ; b=int(input()); p=head ; q=p
while _____:
    q=p
    _____
```

```
if p==head:
    a.append(_____)
```

```
else:
    a.append(_____)
```

4. 在链表 a 中删除数值为 b 的节点:

```
a=[[1, 1], [2, 2], [3, 3], [4, 4], [5, 5], [6, 6], [7, 7], [8, -1]]
head=0 ; p=q=head ; b=int(input("删除节点的值为: "))
while p!=-1 and a[p][0]!=b:
    q=p
    p=a[p][1]
```

```
if p==head:
    _____
```

```
else:
    _____
```

```
p=head
while a[p][1]!=-1:
    print(a[p][0], end="->")
    p=a[p][1]
print(a[p][0])
```

链表遍历1:
1 2 3 4
链表遍历2:
1->2->3->4

```
>>> %Run '删节点.py'
删除节点的值为: 1
2->3->4->5->6->7->8
>>> %Run '删节点.py'
删除节点的值为: 4
1->2->3->5->6->7->8
```

【提高练习】

1. 有如下 Python 程序段:

```
a = [1, 5, 2, 7, 9, 6, 8, 28, 10]; n = len(a); c = [1]*n; p = [-1] * n
for i in range(1, n):
    for j in range(i-1, -1, -1):
        if a[j] < a[i] and c[j] >= c[i]:
            c[i] = c[j] + 1; p[i] = j
            maxlen = max(c)
for i in range(n-1, -1, -1):
    if maxlen == c[i]:
        m = i
tot = 0
while m != -1:
    tot += a[m]; m = p[m]
print(tot)
```

运行该程序段后, 输出的结果是

- A. 27 B. 32 C. 45 D. 50

2. 有如下 Python 程序段:

```
link = [[2, 1], [2, 2], [2, 3], [5, 4], [6, -1]] ; pre = head = 0 ; cur = link[pre][1]
```

```

while cur != -1:
    if link[pre][0] != link[cur][0]:
        (1) _____
    else:
        (2) _____
        (3) _____
p=head
while p != -1:
    print(link[p][0],end=" ")
    p = link[p][1]

```

删除升序链表 link 中重复的数据，仅保留下不重复的数据，上述程序中划线处可选语句为

①cur=link[cur][1] ②pre=cur ③link[pre][1]=link[cur][1]

则(1)、(2)、(3)处正确的语句顺序是

- A. ③②① B. ③①② C. ②①③ D. ②③①

3. 有如下 Python 程序段：

```

a = [3, 2, 4, 1, 3, 2, 5, 3] ;n = len(a);last = [-1] * n ; cur = [-1] * n
for i in range(n):
    last[i] = cur[a[i]]
    cur[a[i]] = i
i = 0; j = 0; ans = 0
while j < n:
    while j < n and last[j] < i:
        j += 1
    if j - i > ans:
        ans = j - i
    i += 1
print(ans)

```

执行该程序段，最后的输出结果是

- A. 3 B. 4 C. 5 D. 6

4. 使用列表 d 模拟链表结构（节点数大于 0），每个节点包含数据区域和指针区域，head 为头指针。现要判断链表是否为回文链表，如 1->2->1 就是回文链表。我们可以先找到中间节点，然后反转后半部分链表，最后逐节点比较前后部分链表节点的值，实现该功能的部分程序段如下，方框中应填入的正确代码为：

```

def func(lst, head):
    p = q = head
    while q != -1 and lst[q][1] != -1:
        p = lst[p][1]
        [ ]
    pre = -1; c = p
    while c != -1:
        [ ]
        a, b = head, pre
    while b != -1:
        if lst[a][0] != lst[b][0]:
            return False
        a, b = lst[a][1], lst[b][1]
    return True
link_list = [[1, 3], [1, 4], [0, 1], [0, -1], [2, 0], [1, 2]]
head = 2
print(func(link_list, head)) # 输出: True

```

①	②
if lst[q][1] == -1: q = lst[q][1] else: q = -1	if lst[q][1] != -1: q = lst[lst[q][1]][1] else: q = -1
③	④
cn = lst[c][1] lst[c][1] = pre pre = c c = cn	c = cn lst[c][1] = pre pre = c cn = lst[c][1]

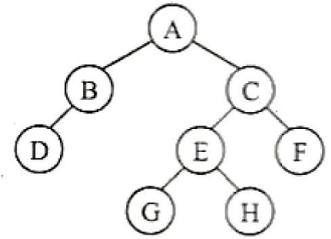
- A. ①③ B. ②③ C. ①④ D. ②④

高三上信息技术 单选提高练习 1

1. 已知某二叉树的中序遍历结果和后序遍历结果分别为 BCAFEFGD 和 CBFEGDA，则该二叉树中没有左孩子节点的节点是
 A. B C F G B. C D F G C. A E F G D. B D F G

2. 有二叉树如图所示，下列关于该二叉树的说法正确的是

- A. 该二叉树共有 7 个节点
- B. 该二叉树共有 4 个叶子节点
- C. G 是 E 的孩子节点, 同时也是 F 的孩子节点
- D. 将 D 从 B 的左孩子节点改为 B 的右孩子节点, 该二叉树的中序遍历结果不变



第 2 题图

3. 假设队列的空间足够, 队首指针 head 和队尾指针 tail 经过“出队、入队、出队、出队、入队、入队、出队”这一系列操作后, head = 7, tail = 9。则操作前 head 和 tail 的值分别为

- A. 11 12 B. 2 5 C. 3 6 D. 3 5

4. 定义如下函数:

```

def fun(n):
    if n==0 or n==1:
        return 1
    else:
        return fun(n-1)*2+fun(n-2)*3
    
```

下列关于该程序段的说法, 正确的是

- A. 该程序段运用了迭代的思想
- B. 当输入值 n 为 3 时, 函数 fun 一共被调用了 4 次
- C. 当输入值 n 为 5 时, 函数 fun 一共被调用了 15 次
- D. 当输入值 n 为 4 时, 函数 fun 的返回值为 49

5. 有如下 Python 程序段:

```

n = int(input())
k = 0
for i in range(n):
    for j in range(i + 1, n + 2):
        k = k + 1
    
```

该程序段的时间复杂度为

- A. $O(n)$ B. $O(n^2)$ C. $O(2^n)$ D. $O(n/2)$

6. 有如下 Python 程序段:

```

import random
p="abcde*" ; st=[] ; s="" ; i=0
while i<=5:
    m=random.randint (0,1)
    if m==0:
        st.append (p[i])
        i+=1
    elif len(st)>0:
        s+=st.pop ()
    
```

执行该程序段后, s 的结果可能是

- A. a* B. cdabe C. abcde* D. cdba

7. 有如下 Python 程序段:

```

def search (arr,n):
    left=0
    right=len(arr)-1
    while left<=right:
        m=(left+right)//2
        if arr[m]==n:
            return arr [m]
        elif arr [m]<n:
            left=m+1
        else:
            right=m-1
    return arr[:right+1]
print (search ([2, 6, 15, 21, 23, 33, 76, 88, 91], 24))
    
```

执行该程序段后, 输出的结果是

- A. [] B. [2, 6, 15, 21, 23] C. [2, 6, 15, 21, 23, 33] D. [2, 6, 15, 21, 23, 33, 76, 88, 91]

8. 有如下 Python 程序段:

```
def bubbleSort (n):
    if n==1:
        return
    for i in range (n-1):
        if arr[i]>arr[i+1]:
            arr[i],arr [i+1]]=arr[i+1],arr[i]
    bubbleSort(n-1)
```

from random import randint

n=randint(3,5)

bubbleSort(n)

若数组 arr 的值为[64, 34, 25, 12, 22, 11, 90], 则调用函数 bubbleSort(n)后, arr[3]的值不可能是

- A. 12 B. 25 C. 34 D. 64

9. 定义如下函数:

```
def jc(n):
    if n==1: #①
        return n
    return n*jc(n-1) #②
```

执行语句 x=jc(5)后, 下列说法正确的是

- A. x 的计算结果为 120 B. 程序执行完毕, ①处代码共执行 1 次
C. 程序执行完毕, ②处代码共执行 5 次 D. 若①处代码改为 n<2, 程序将无法正常运行

10. 列表 a 包含 8 个互不相等的元素, 即 a[0], a[1], a[2], ..., a[7], 有如下 Python 程序段:

```
cnt1=0;cnt2=0
for i in range(len(a)):
    for j in range(1,len(a)-i):
        if a[j-1]>a[j]:
            a[j-1],a[j]=a[j],a[j-1]
            cnt2+=1
    cnt1+=1
```

已知 a 的值为[3, 8, 2, 4, 7, 5, 1, 6], 执行该程序段后, cnt1*cnt2 的值是

- A. 168 B. 224 C. 392 D. 420

11. 有如下 Python 程序段:

```
import random
a=[0]*10
k=random.randint(1,5)
a[0]=k;n=9
for i in range(1,n+1):
    a[i]=a[i-1]+random.randint(1,5)*2
for i in range(n,k,-1):
    a[i]=a[i-1]
a[k]=k
```

执行该程序段后, 列表 a 的值可能是

- A. [2, 6, 2, 10, 17, 20, 22, 26, 34, 40] B. [5, 7, 13, 19, 21, 5, 23, 25, 31, 33]
C. [1, 3, 7, 11, 21, 27, 1, 37, 43, 45] D. [4, 10, 16, 4, 20, 24, 34, 40, 44, 50]

12. 小明在学习了随机数模块后, 编写了如下 Python 程序段:

```
import random
n=int(input("输入 n 的值:"))
st=[0]*n
head,tail=0,len(st)-1
p=random.randint(5,8)*10+random.randint(0,9)
while head<=tail:
    p=p+random.randint(1,3)
    if p%2==0:
        st[head]=p
        head+=1
    else:
        st[tail]=p
        tail-=1
print(st)
```

若输入 n 的值为 5, 则执行该程序段后, 输出的结果可能是

- A. [48, 84, 81, 79, 77] B. [88, 92, 95, 91, 87] C. [76, 77, 82, 84, 79] D. [80, 81, 87, 85, 83]