

青少年软件编程（Python）等级考试试卷（四级）

分数：100 题数：38

一、单选题(共 25 题，每题 2 分，共 50 分)

1. 下列不是评判一个算法优劣的标准是？（ ）
 - A. 时间复杂度
 - B. 空间复杂度
 - C. 难易度
 - D. 健壮性
2. 某自定义函数中使用了 3 个变量,其中 2 个变量都指定了默认值,调用该函数时,参数的个数最少为几个？（ ）
 - A. 0
 - B. 2
 - C. 3
 - D. 1
3. Python 中用于字符与其 ASCII 码相互转换的内置函数是？（ ）
 - A. float() abs()
 - B. pow() format()
 - C. ord() chr()
 - D. round() import()
4. Python 中，可以将数字类字符串转换成整数的函数是？（ ）
 - A. int()
 - B. float()
 - C. bool()
 - D. class()
5. 运行下列这段程序，正确的输出结果是？（ ）

```
LS = ["apple", "red", "orange"]
def funb(a):
    LS.append(a)
funb("yellow")
print(LS)
```

 - A. ["apple","red","orange"]
 - B. ["apple","red","orange","yellow"]
 - C. []
 - D. ["yellow"]
6. 下列代码的输出结果是？（ ）

```
def func(a, b):
```

```
c=a**2+b
b=a
return c
a=10
b=2
c=func(b,a)+a
print(c,b)
```

- A. 102 2
- B. 24 10
- C. 24 2
- D. 102 10

7. 下列关于函数返回值的描述正确的是？（ ）

- A. 函数可以没有返回值，也可以有一个或多个返回值
- B. 函数定义中最多含有一个 return 语句
- C. 在函数定义中用 return 语句时，至少给一个返回值
- D. 函数中 return 语句只能放在函数定义的最后

8. 运行下列这段程序，正确的输出结果是？（ ）

```
def reverseA(s):
    t=""
    for i in range(0,len(s)):
        t=s[i]+t
    return t
print(reverseA("reverse"))
```

- A. esrever
- B. reverse
- C. rsereve
- D. rseveer

9. 下列导入 pandas 模块的方法不正确的是？（ ）

- A. import pandas
- B. import pandas as p
- C. import pandas *
- D. from pandas import *

10. Python 中提供了大量的标准库模块，比如利用 time 模块可以获取计算机的时钟信息，包括日期和时间，若想

要实现延时 1 分钟，则应该使用的语句是？（ ）

- A. sleep(3600)
- B. time.time(60)
- C. time.asctime(1)
- D. time.sleep(60)

11. 使用分治算法的基本步骤是？（ ）
- A. 分解、解决、合并
 - B. 分解、解决
 - C. 合并、解决
 - D. 合并、解决、分解
12. 二分查找法是运用哪种策略实现的算法？（ ）
- A. 分治法
 - B. 贪心法
 - C. 回溯法
 - D. 动态规划法
13. 下列程序是分治算法的典型应用，其运行结果是？（ ）

```
def dividAndConquer(arr,left,right):  
    if (right == left + 1) or (right == left):  
        return max(arr[left],arr[right])  
    mid = int((left + right) / 2)  
    leftMax = dividAndConquer(arr,left,mid)  
    rightMax = dividAndConquer(arr,mid,right)  
    return max(leftMax,rightMax)  
arr1 = [8, 1, 14, 19, 5]  
print(dividAndConquer(arr1,0,4))
```

- A. 1
- B. 19
- C. 8
- D. 5

14. 下列程序输出的结果是？（ ）

```
def change(a,b):  
    a=10  
    b+=a  
a=4  
b=5  
change(a,b)  
print(a,b)
```

- A. 4 5
- B. 10 5
- C. 4 15
- D. 10 15

15. 下列程序运行的结果是？（ ）

```
def a():  
    print("here")  
b()  
a()  
def b():  
    print("there")  
a()  
b()
```

- A. 出错
- B. 死循环
- C. here
- D. here
there

16. 设函数 f 定义如下：

```
def f(x,y = 0):  
    pass
```

下列选项执行时，出现错误的是？（ ）

- A. f(,2)
- B. f(1,2)
- C. f(1)
- D. f(1,)

17. 下列程序段运行的结果是？（ ）

```
def check(x = 2,y = 3):  
    print(x,y)  
check(y=6,x=5)
```

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

18. def AddApple(fruit=None):

```
    if fruit is None:  
        fruit=[]  
    fruit.append('Apple')  
    return fruit  
AddApple()  
AddApple()  
print(AddApple(['Pear']))
```

以上程序段运行的结果是？（ ）

- A. ['Pear','Apple']
- B. ['Pear','Apple','Apple']
- C. ['Apple','Pear','Apple','Apple']
- D. ['Apple']

19. def ThankYou(name):

```
    print("感谢%s 对公司的贡献！" %name)
```

ThankYou('张三')

ThankYou()

以上程序运行的结果是？（ ）

打印输出：感谢张三对公司的贡献！

A.

然后报错：TypeError: ThankYou() missing 1 required positional argument: 'name'

打印输出：

B.

感谢张三对公司的贡献！

感谢对公司的贡献！

C.

无法输出。

D.

需要增加 main 函数才能正确运行。

20. 编写程序计算 $1+1/2+1/3+.....+1/n$ 的结果，可以使用哪种调用函数自身的算法？（ ）

A. 枚举

B. 递归

C. 解析

D. 分治

21. 用下面的程序求解计算 $s=1+3+5+7+9$ 的值，请选择横线处应填写的代码？（ ）

```
def Sum(n):
```

```
    if n<=1:
```

```
        return 1
```

```
    else:
```

```
        return _____
```

```
print(Sum(9))
```

A. $n+Sum(n-1)$

B. $n+Sum(n+1)$

C. $n+Sum(n+2)$

D. $n+Sum(n-2)$

22. 下列选项中，哪一项不是递归函数必须要具备的条件？（ ）

A. 明确的边界条件

- B. 边界值
- C. 循环语句
- D. 终止条件

23. 用递归算法实现 5 的阶乘，则下方的程序中横线上需要的代码是？（ ）

```
def func(n):  
    if n<_____  
        return func(n+1)*n  
    else:  
        return 1  
print(func(1))
```

- A. 4
- B. 5
- C. 6
- D. 7

24. 执行以下代码，程序的输出结果是？（ ）

```
def weight(n):  
    if n==1:  
        return 100  
    else:  
        return weight(n-1) +10  
print(weight(3))
```

- A. 100
- B. 110
- C. 120
- D. 130

25. 以下关于递归与递推的说法，错误的是？（ ）

- A. 递归算法不涉及高深的数学知识，比较容易理解。
- B. 递归过程一般通过函数或子过程来实现。
- C. 递归算法是递推分解问题，然后再将最简单情况的解回归成大问题的解。
- D. 存在既可以用递归算法解决，也可以用递推算法解决的问题。

二、判断题(共 10 题，共 20 分)

26. 函数中 return 语句只能放在函数定义的最后。（ ）

正确 错误

27. Python 定义函数时如果没有参数，也不能省略后面的空括号。（ ）
- 正确 错误
28. 函数的参数分为形参和实参两种，函数调用中实参和形参的值可以互相传递。（ ）
- 正确 错误
29. 导入模块中指定成员时，可采用的语句是：import 成员名 1，成员名 2 from 模块名。（ ）
- 正确 错误
30. 二分搜索、快速排序可以使用分治算法进行求解。（ ）
- 正确 错误
31. 调用带有默认值参数的函数时，不能为默认值参数传递任何值，只能使用函数定义时设定的默认值。（ ）
- 正确 错误
32. 在 Python 中，可以在自定义函数中再定义函数。（ ）
- 正确 错误
33. 在定义一个函数时，如果出现调用自身的成分，则称为递推。（ ）
- 正确 错误
34. 递归算法通常显得很简洁，因为多次调用自身，所以运行效率较高，应该大力提倡用递归算法设计程序。
- （ ）
- 正确 错误
35. 同样的问题使用不同的算法编程解决，其时间复杂度可能不一样。（ ）
- 正确 错误

三、编程题(共 3 题，共 30 分)

36. 验证哥德巴赫猜想：任意一个大于等于 4 的偶数总可以分解为两个素数之和。请在下面的横线上补全代码，

使得通过键盘输入一个偶数 $x(x < 1000)$ ，输出这两个素数。如果有多组解，只需要输出一组，并且第一个素数要尽可能地小。例如，输入“10”，输出“3 7”；输入“18”，输出“5 13”。

```
def prime(x):
    for i in range(2, x):
        if x%i == 0:
            return ①
    return 1

x=int(input("输入一个偶数 x:"))

for k in range( ② 2 , x//2+1):
    if prime(k) and prime(③):
        print(④)
        break
```

试题难度：一般

试题解析：

评分标准：

- (1) 0 或等效答案 ； (3 分)
- (2) 2 或等效答案 ； (3 分)
- (3) $x-k$ 或等效答案 ； (3 分)
- (4) $k, x-k$ 或等效答案 。 (3 分)

37. 回文数是指正序（从左向右）和倒序（从右向左）读都是一样的整数，例如 12321、765567 就是回文数。请

补全下列代码，编程实现用分治的方法判断一个整数是否是回文数。

```
def ispel(s):
    num = len( s ① )
    if num == 0:
        return False
    if num == 1:
        return True
    if num == 2:
        return s[0] == s[num-1] ②
    else:
        return s[0] == s[num - 1] and ispel(s[1: ③])

st = input("请输入一个整数：")

if ④ ispel(st) == True:
    print("输入的是回文数")
else:
```

```
print("输入的不是回文数")
```

试题难度：一般

试题解析：

评分标准：

- ① s 或等效答案； (2 分)
- ② s[1]或等效答案； (2 分)
- ③ num-1 或等效答案； (2 分)
- ④ ispel(st) 或等效答案。 (2 分)

38. 小青蛙爬楼梯。楼梯有 n 级。小青蛙每次只能爬 1 级或 2 级。

当楼梯只有 1 级时，青蛙只有 1 种爬法。

当楼梯有 2 级时，青蛙有 2 种爬法：一次爬 2 级；一次爬 1 级，爬 2 次。

请从键盘上接收楼梯级数 n ，输出青蛙有几种爬法。

示例输入

请输入楼梯级数：8

示例输出

8 级楼梯共有 34 种爬法。

程序如下：

```
def Climb(n):  
    if n==1 ①:  
        return 1  
    elif n==2:  
        return 2 ②  
    else:  
        return ③ Climb(n-1)+Climb(n-2)  
  
n=int(input("请输入楼梯级数："))  
if n>0:  
    print("%d 级楼梯共有%d 种爬法。"%(④ n, ⑤d))
```

试题难度：容易

试题解析：

评分标准：

(1) `n==1;` (2 分)

(2) `return 2;` (2 分)

(3) `Climb(n-1)+Climb(n-2);` (2 分)

(4) `n;` (2 分)

(5) `Climb(n).` (2 分)