

分数：100.00 题数：38

1. 对于数列 3, 8, 11, 15, 17, 19, 25, 30, 44, 采用“二分查找”法查找 8, 需要查找多少次?

试题解析：按二分查找法的规律，每次先查找中间值，进行比较。

试题解析：pip 工具中没有 clear 方法。

试题解析：将参数 3 传递给匿名函数 f，返回值为 5，故选 C

```
def inverse(s,n=0):         while s:
          n = n * 10 + s % 10
          s = s // 10
     return n
print(inverse(456,123))
```

- A、654123
- B、123456
- C、123654
- D、654321

题型：单选题

答案：C

难度：一般

试题解析：调用函数 `inverse(456,123)`，将 456 逐位取出，并累加到 123 的后面，故选 C

5. 下列有关循环和递归的描述正确的是？（ ）

- A、递归思想代码清晰简洁，可读性强
- B、递归代码中不能有循环结构的语句
- C、递归是从问题的起点出发，逐渐将复杂问题化为简单问题，最终求得问题
- D、能用递归实现的，一定能用循环代码实现

题型：单选题

答案：A

难度：一般

试题解析：递归是从问题的目标出发，逐渐将复杂问题化为简单问题，最终求得问题

6. 以下有关 Python 函数的定义表述中错误的是？（ ）

- A、函数的定义必须在主程序调用语句之前出现
- B、在形参列表中必须先列出有默认值的形参，再列出没有默认值的形参
- C、实参是实际占用内存地址的，而形参不占用
- D、`def` 关键字后面加函数名定义函数，定义必须以冒号结尾

题型：单选题

答案：B

难度：一般

试题解析：在形参列表中必须先列出没有默认值的形参，再列出有默认值的形参。

7. 如下代码运行后下面选项中描述错误的是？（ ）

```
def pph(a,b):  
    &nbsp;&nbsp;&nbsp;c=a**2+b  
    &nbsp;&nbsp;&b=a  
    &nbsp;&nbsp;&return c  
a=10  
b=100  
c=pph(a,b)+a  
print(a,' ',b,' ',c)
```

- A、执行该函数后，变量 a 的值为 10
- B、执行该函数后，变量 b 的值为 100
- C、执行该函数后，变量 c 的值为 200
- D、该函数名称为 pph

题型：单选题

答案：C

试题解析：函数基本概念

```
def fibona(x):  
         if x==1 or x==2:  
              f=1  
         for i in range(3,x+1):  
              f=fibona(x-1)+fibona(x-2)  
         return f  
n=int(input("请输入数列第几项 : "))  
m=fibona(n)  
print("数列的第"+str(n)+"项的值为"+str(m))
```

- 难度：容易

[illegible]

- 难度：一般

```
x = 3
def calc():
    x = 5
print(calc())
```

- A、3
B、5
C、无输出

试题解析：自定义函数没有 return 语句，返回值为 None，故选 D

试题解析：A。问题的可分性和解的可归并性是应用分治算法的前提。

答案：B

难度：一般

试题解析：每次弹起的高度都是上次的一半，因此递归式为 $\text{height}(n-1)/2$

13. 有如下 Python 程序，执行该程序后，结果是？（ ）

```
g = lambda x,y=3,z=5:x+y+z
print(g(2))
```

- A、 2
- B、 5
- C、 7
- D、 10

题型：单选题

答案：D

难度：一般

试题解析：匿名函数 g 的功能是将 3 个参数 x,y,z 之和，故选 D

14.

下面的程序输出 1~100 之间能被 7 整除但不能同时被 5 整除的所有整数。

```
k=1
while k<101:
    &nbsp; &nbsp; &nbsp; &nbsp; if k%7==0 and k%5 !=0:
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; print(k)
    &nbsp; &nbsp; &nbsp; &nbsp; k += 1
```

根据下面哪个选项的方法优化后，程序的运行效率最高？（ ）

- A、 将 k=1 改为 k=7
- B、 将 k
- C、 将 k += 1 改为 k += 7
- D、 将 k=1 改为 k=7，同时将 k += 1 改为 k += 7

题型：单选题

答案：D

难度：一般

试题解析：选项 A 可以少执行 6 次，选项 B 可以少执行 2 次，选项 D 可以少执行 86 次。选项 C 程序不能实现设计的功能。

15. 下列程序段的运行结果为？（ ）

```
def f(n):
    &nbsp; &nbsp; &nbsp; &nbsp; if n<=1:
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; return 1
    &nbsp; &nbsp; &nbsp; &nbsp; else:
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; return f(n-1)*3
print(f(5))
```

- A、 9
- B、 27
- C、 81
- D、 243

题型：单选题

答案：C

难度：容易

试题解析： $f(5)=f(4)*3=f(3)*3*3=f(2)*3*3*3=f(1)*3*3*3*3=1*81=81$

16. 下列选项中，关于如何安装第三方库的说法正确的是？（）

A、如果电脑没有联网，仍然可以使用 pip 工具安装本地的 whl 文件

B、必须使用命令行工具安装第三方库

C、第三方库只要可以用 pip 完整的下载下来，就可以成功安装

D、安裝上 Anaconda 就安裝了所有的第三方庫

题型：单选题

答案：A

难度：一般

试题解析：可执行文件不需要使用命令行工具就可以安装。没有编译的第三方库，即使能够下载，也无法安装。Anaconda 里面包含了 800 多个第三方库，但是不是全部的第三方库。

17.

运行以下程序输出的结果是？（ ）

 $y=2$

```
def fun():    
```

 global y

$$y=1$$

```

    &nbsp; &nbsp; &nbsp; print(y)

```

fun()

```
print(y)
```

A.

2

1

B、

2

2

C、

1

2

D,

1

1

题型：单选题

答案：D

难度：较难

试题解析：Python 中定义函数时，若想在函数内部对函数外的变量进行操作，就需要在函数内部声明其为 global 以改变它的值。

18. 下面哪种算法使用了分治的方法？（ ）

A、插入排序

B、快速排序

A, $c(n-1)+2*(n-1)$

- B、 $c(n-1)+c(n-2)$
C、 $c(n-1)+2*n$
D、 $c(n-1)+2*(n+1)$

题型：单选题

答案：A

难度：较难

试题解析：

设这 n 个圆将平面分成 a_n 个区域。易知， $a_1=2, a_2=4$ 。现在假设前 $n-1$ 个圆将平面分成了 a_{n-1} 个区域，当加入第 n 个圆(虚线圆)时，由题设这个圆与前面的 $n-1$ 个圆一定交于 $2(n-1)$ 个点，这 $2(n-1)$ 个点把第 n 个圆分成 $2(n-1)$ 条弧，而每条弧正好将前面的 $n-1$ 个圆分成的区域中的其经过的每个区域分成 2 个区域，故新加入的第 n 个圆使所成的区域数增加了 $2(n-1)$ 。因此可以建立如下带初值的递推关系：

$$\begin{aligned} a_n &= a_{n-1} + 2(n-1) \\ a_1 &= 2 \end{aligned}$$

22. 有如下 Python 程序段，执行该程序后，结果是？ ()

```
def fun(*p):  
    &nbsp; &nbsp; &nbsp; return sum(p)  
print(fun(1,3,5))
```

- A、4
B、6
C、8
D、9

题型：单选题

答案：D

难度：一般

试题解析：该函数功能是将参数 1,3,5 求和，故选 D

23. 以下关于全局变量和局部变量的表述正确的是？ ()

- A、如果在函数中定义的局部变量与全局变量同名，则全局变量屏蔽局部变量
B、可以通过 global 关键字，通过全局变量修改局部变量
C、nonlocal 关键字用来在函数或局部作用域使用内层（非全局）变量
D、全局变量的作用域一定比局部变量的作用域大

题型：单选题

答案：D

难度：一般

试题解析：如果在函数中定义的局部变量与全局变量同名，则局部变量屏蔽全局变量。可以通过 global 关键字，通过局部变量修改全局变量。nonlocal 关键字用来在函数或局部作用域使用外层（非全局）变量。局部变量的作用域是指程序内部，全局变量的作用域是整个程序。

24.

关于以下程序，下列表述中错误的一项是？ ()

```
c=1  
def fun(n):  
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; a=1  
    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; for b in range(1,n):
```

A、c 是全局变量，a 是局部变量
B、n 是形式参数，当 n=5 时，程序输出 120 1
C、程序实现求阶乘
D、range()函数是 python 内置函数

试题解析：fun(n)函数的功能是求阶乘，range(1,n)是从1循环到n-1。

试题解析：关键字实参

试题解析：

29. 执行以下代码：

```
def fun( mylist ):
    &nbsp; &nbsp; mylist.append([1,2,3,4])
    &nbsp; &nbsp; print("函数内取值: ", mylist)
    &nbsp; &nbsp; return
mylist = [5,6,7]
fun( mylist )
print("函数外取值: ", mylist)
```

程序输出的结果为：函数内取值: [5, 6, 7, [1, 2, 3, 4]]

 函数外取值: [5, 6, 7, [1, 2, 3, 4]]。

题型：判断题

答案：正确

难度：一般

试题解析：实例中传入函数的和在末尾添加新内容的对象用的是同一个引用，输出是一样的。

30. 定义 Python 函数时，如果函数中没有 return 语句，则该函数返回值是 None。

题型：判断题

答案：正确

难度：一般

试题解析：定义 Python 函数时，如果函数中没有 return 语句，则该函数返回值 None

31. 执行以下代码：

```
sum=0
def fun(arg1,arg2):
    &nbsp; &nbsp; sum=arg1+arg2
    &nbsp; &nbsp; print(sum)
    &nbsp; &nbsp; return sum
fun(5,10)
print(sum)
```

程序输出的结果为：15

 15

题型：判断题

答案：错误

难度：一般

试题解析：第 1 个 print(sum)是函数内的局部变量，输出 15。第 2 个 print(sum)是函数外的全局变量，输出 0。

32. 对于一个复杂问题，如果所分解出的各个子问题之间相互不独立，则不适合使用分治算法。

题型：判断题

答案：正确

难度：一般

试题解析：正确。分治算法要求子问题是相互独立的

33. 执行以下代码：

```
def fun( name, age = 30 ):
    &nbsp; &nbsp; &nbsp; print("Name:", name)
    &nbsp; &nbsp; &nbsp; print("Age:", age)
    &nbsp; &nbsp; &nbsp; return
fun( age=40, name="summy" )
fun( name="summy" )
```

程序输出的结果为：

Name: summy Age: 40
Name: summy

Age: 40。

题型：判断题

答案：错误

难度：一般

试题解析：调用函数时，默认参数的值如果没有传入，则被认为是默认值。第二次调用 fun() 时 age 的值没有传入，则默认是 30。

34. 下列程序段运行后的结果是 2。

```
def change(a,b):
    &nbsp; &nbsp; &nbsp; a,b=b,a
    &nbsp; &nbsp; &nbsp; return a
a=2
b=3
print(change(change(a,b),a))
```

题型：判断题

答案：正确

难度：容易

试题解析：自定义函数 change (a,b) 的作用是交换变量 a,b 的值并返回变量 a 的值，change(2,3) 返回的值为 3，change(3,2) 返回的值为 2。

35. 对于斐波那契数列：1，1，2，3，5，……，我们只能采用迭代公式以递推的方式求解。

题型：判断题

答案：错误

难度：一般

试题解析：斐波那契数列可以用多种方式求解。

三、编程题 (共 3 题，共 30 分)

36.

在编写抽奖程序时，为了保证一个人只有一次中奖机会，要检查新抽出来的数字是不是已经被抽中过了。

一种办法是将已经中过奖的人员编号存放在 test_list 里面，然后每抽出一个新的人员编号，判断它是否在中奖人员列表中。

如果没有在中奖人员列表中，说明中奖号码有效，并将它保存进中奖人员列表；如果已经在里面了，就再生成一个新的人员编号。

请你补全下面的代码，实现判断一个数字是否在列表中的功能。

```
#子问题算法（子问题规模为1）
def is_in_list(init_list,num):
             if init_list[0] == num:
                      return True
             else:
                      return False
```

[illegible]

```

    #递归，分治，合并
    res=find_out(left_list,num) ③ find_out(right_list,num)

```

```

    &nbsp; &nbsp; &nbsp;return res

```

```
if __name__ == "__main__":  
        #测试数据  
        test_list = [18,43,21,3,28,2,46,25,32,40,14,36]  
        #查找  
        print(____④____)    
```

程序运行结果：
>>>True

题型：编程题
答案：

参考答案：

(1) `n==1`; 或其他正确答案; (2 分)
(2) `init_list[:n//2], init_list[n//2:]`; 或其他正确答案;
(2 分)

(3) or 或其他正确答案 ; (2 分)

(4) find_out(test_list,25) 或其他正确答案。 (2 分)

难度：较难

试题解析：

参考程序：

```
#子问题算法（子问题规模为1）
def is_in_list(init_list,num):
    if init_list[0] == num:
        return True
    else:
        return False

#分治法
def find_out(init_list,num):
    n = len(init_list)
    if n == 1:
        #如果问题的规模等于1，直接解决
        return is_in_list(init_list,num)
    #分解（子问题规模为n/2）
    left_list,right_list = init_list[:n//2],init_list[n//2:]

    #递归（树），分治，合并
    res=find_out(left_list,num) or find_out(right_list,num)

    return res

if __name__ == "__main__":
    #测试数据
    test_list = [18,43,21,3,28,2,46,25,32,40,14,36]
    #查找
    print(find_out(test_list,25)) #True
```

37.

乘法运算等于多个加法运算的和。比如， 3×2 可以理解为 $3+3$ ，也可以理解为 $2+2+2$ 。下面的程序使用递归算法演示了计算两个自然数的乘积的过程。请你补全代码。


```
mht = my abs(x1-x2)+my abs(y1-y2) &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
```

```

&nbsp; &nbsp; &nbsp; #方法 1
    mht&nbsp;=&nbsp;my_abs(y1-y2)+my_abs(x1-x2) &nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; &nbsp; #方法 2
    mht&nbsp;=&nbsp;sum([my_abs(x1-x2),my_abs(y1-y2)]) &nbsp; &nbsp; &nbsp;
&nbsp; #方法 3
    mht&nbsp;=&nbsp;sum((my_abs(x1-x2),my_abs(y1-y2))) &nbsp; &nbsp; &nbsp;
&nbsp; #方法 4
    #计算路上骑行时间
    time_on_the_road = mht/1000/v*60 &nbsp; &nbsp; &nbsp; #方法 1
    time_on_the_road = mht/(1000*v)*60 &nbsp; &nbsp; &nbsp; #方法 2
    time_on_the_road = mht*60/1000/v &nbsp; &nbsp; &nbsp; #方法 3
    time_on_the_road = (mht*60)/(1000*v) &nbsp; &nbsp; &nbsp; #方法 4
    print('A、 B 两点的曼哈顿距离为{ }米'.format(mht))
    print('预计路上骑行时间需要{ }分钟'.format(time_on_the_road))

```