

北京工业大学 2021–2022 学年 秋季学期 Python 编程 (A 卷解析)

2022 年 1 月 8 日 14:30-16:30 班级 _____ 姓名 _____ 学号 _____

(注: 选择、判断、填空题的解答直接写在这张试题纸上, 该试题纸一并上交。其他题目写在答题本上)

一、判断题 (每小题 2 分, 共 20 分) 在题目中的括号内标出✓或✗.

- 1. 函数中 else 关键字的逻辑作用与 if/else 中 else 关键字的逻辑作用一致. 【X】
 - 2. 已知 $x = 3$, 那么执行语句 $x += 6$ 之后, x 的内存地址不变. 【✓】
 - 3. 正则表达式对象的 match() 方法可以在字符串的指定位置开始进行指定模式的匹配. 【✓】
 - 4. Python 变量使用前必须先声明, 并且一旦声明就不能再当前作用域内改变其类型. 【X】
 - 5. Python 支持多继承, 如果不同父类中有相同的方法名, 而子类中调用时没有指定父类名, 则 Python 解释器将从左向右按顺序进行搜索. 【✓】
 - 6. 字符串属于 Python 有序序列, 和列表、元组一样都支持双向索引. 【✓】
 - 7. 扩展库 os 中的方法 remove() 可以删除带有只读属性的文件. 【X】
 - 8. 使用内置函数 open() 且以”w” 模式打开的文件, 文件指针默认指向文件尾. 【X】
 - 9. Python 中的列表被切片后, 列表中的某个元素被删除后别的元素将不再自动收缩补齐空格. 【X】
 - 10. 相同内容的字符串使用不同的编码格式进行编码得到的结果并不相同. 【✓】

二、单选题 (每小题 2 分, 共 20 分) 在下列各小题中选择其中一个答案, 标注在题目中的括号内.


```
a = "Python 等级考试"
```

```
a = "Python 等级考试"  
b = "="  
c = ">"  
print("{0:{1}{3}{2}}".format(a, b, 25, c))
```

- (A) Python 等级考试 =====
 - (B) »»»»>Python 等级考试

- (A) showNnumber([2, 4, 5])
(B) showNnumber('abcesf')
(C) showNnumber(3.4)
(D) showNumber((12, 4, 5))

10. 函数如下，打印结果哪项是正确的

【 B 】

```
def changeInt(number2):
    number2 = number2 + 1
    print("changeInt: number2=", number2)
# 调用
number1 = 2
changeInt(number1)
print("number:", number1)
```

- (A) changeInt: number2= 3 number: 3 (B) changeInt: number2= 3 number: 2
(C) number: 2 changeInt: number2= 2 (D) number: 2 changeInt: number2= 3

三、填空题 (每小题 2 分, 共 20 分) .

- 已知列表 $x = [1, 2]$, 执行语句 $y = x[:]$ 后, 表达式 $\text{id}(x) == \text{id}(y)$ 的值为 False。
- 表达式 $[i \text{ for } i \text{ in range}(10) \text{ if } i > 8]$ 的值为 [9, 9]。
- 已知列表 $x = [] * 4$, 那么执行语句 $x[1].append(3)$ 之后, x 的值为 [3], [3], [3], [3]。
- 已知 $mlist = [1, 'a', 3, 4, 5]$, 并且有 $result = map(lambda x: x * 2, mlist)$, 那么表达式 $\text{print(list(result))}$ 的值是 [2, 'aa', 6, 8, 10]。
- 正则表达式模块 re 的 search 方法用来在整个字符串中进行指定模式的匹配。
- 以下程序运行结果为 15 :

```
fruits = {'apple': 3, 'banana': 4, 'pear': 5}
fruits['banana'] = 7
print(sum(fruits.values()))
```

7. 假设有 Python 程序文件 demo.py, 代码如下:

```
def main():
    if __name__ == '__main__':
        print(1)
    else:
        print(2)

main()
```

将该程序文件直接运行时输出结果为 1，作为模块导入时得到结果 2。

8. 表达式 `list(map(lambda x: len(x), ['a', 'bb', 'ccc']))` 的值为 [1, 2, 3]。

9. 以下程序的输出结果是 [1, 2, 3]：

```
a = [[1], 2, 3]
b = a
a[0] = 1
print(b)
```

10. Python 中，如果我们只需要计算平方根的函数，为了能提高计算效率，我们在引入 `math` 库的时候，最优库函数引入方式的编程应该为 `from math import sqrt`。

四、程序分析题 (每空 2 分，共 10 分)

1. (10 分) 将函数补充完整，使其具有以下功能：给定一个整数数组 `nums` 和一个目标值 `target`，请你在该数组中找出和为目标值的两个整数，并返回他们的数组下标。

```
def twoSum(nums, target):
    for i in range(len(nums)):
        for j in range(i, len(nums)):
            -----
            return i, j
    return False
```

解

- 关键代码：`if nums[i] + nums[j] == target and i != j:` 2 分

□

2. (10 分) 下列程序的输出结果为____：

```
def demo(i):
    if i == 5:
        return 6
    else:
        return (demo(i+1) - 8 + i) * 2

i = 2
print(demo(i))
```

解

- 答案: -16 2 分

□

3. (10 分) 托运货物收费是根据货物重量来算的, 某托运处的收费标准是: 货物重量在 50 千克(包括)以下的, 每千克 0.5 元, 超过 50 千克部分每千克 0.6 元。写一程序, 输入货物重量, 输出收费金额。请补充下列程序空缺的部分:

```
def fee():
    weight = input("请输入货物的重量, 以千克为单位")
    if eval(weight) <= 50 and eval(weight) >= 0:
        money = -----
    elif eval(weight) > 50:
        money = -----
    else:
        print("您的输入有误, 请重新输入")
    print("收取金额为: {}".format(money))

fee()
```

解

- 第一个空: eval(weight) * 0.5 2 分
- 第二个空: 50 * 0.5 + (eval(weight) - 50) * 0.6 2 分

□

4. (10 分) 下列程序表示计算员工工资, 工资与工龄有关, 计算公式为 ‘salary = 3000 + 100 * age‘, 输出更改前后的工资。现有一员工 ‘jack‘, 原始工龄为 4, 现在工龄为 5, 请补充程序空缺:

```
class Worker:
    def __init__(self, name, workAge):
        self.name = name
        self.workAge = workAge

    def salary(self):
        return -----
    salary = property(salary)

    def dispSalary(self):
        print(self.salary)
```

解

- 答案: $3000 + 100 * \text{self.workAge}$ 2 分

□

五、编程题 (每小题 10 分, 共 30 分) . 1. (10 分) 函数形式实现将 40 插入到列表 ‘mlist = [34, 86, 18, 50]‘ 中, 并完成插入排序。插入排序的基本操作为: 将待排序元素插入到已排序序列中的正确位置, 同时将已排序序列中的元素后移, 以便为新元素腾出位置。请补充下列程序的空缺部分, 使其完成上述功能。

```
def insertion_sort(nums, target):  
    # 插入目标值  
    nums.append(target)  
    # 从第二个元素开始遍历  
    for i in range(1, len(nums)):  
        key = nums[i]  
        j = i - 1  
        # 将大于 key 的元素移动到下一位置  
        while j >= 0 and nums[j] > key:  
            nums[j + 1] = nums[j]  
            j -= 1  
        # 将 key 插入正确位置  
        nums[j + 1] = key  
    return nums
```

```
# 示例  
mlist = [34, 86, 18, 50]  
target = 40  
sorted_list = insertion_sort(mlist, target)  
print(sorted_list)
```

2. (10 分) 斐波那契数, 通常用 ‘F(n)‘ 表示, 形成的序列称为斐波那契数列。该数列由 0 和 1 开始, 后面的每一项数字都是前面两项数字的和。也就是:

- ‘ $F(0) = 0$ ‘
- ‘ $F(1) = 1$ ‘
- ‘ $F(N) = F(N - 1) + F(N - 2)$ ‘, 其中 ‘ $N > 1$ ‘。

请用函数形式表示 ‘F(N)‘ 并实现此函数。

- 递归实现：

```
def F(N):  
    if N == 0:  
        return 0  
    elif N == 1:  
        return 1  
    else:  
        return F(N - 1) + F(N - 2)
```

- 迭代实现（更高效）：

```
def F_iterative(N):  
    if N == 0:  
        return 0  
    elif N == 1:  
        return 1  
    a, b = 0, 1  
    for _ in range(2, N + 1):  
        a, b = b, a + b  
    return b
```

3. (10 分) 给你一个数组 ‘arr‘，请你将每个元素用它右边最大的元素替换。如果是最后一个元素，用 -1 替换。完成所有替换操作后，请你返回这个数组。

示例：

- 输入：‘arr = [7, 8, 5, 4, 6, 3]‘
- 输出：‘[8, 6, 6, 6, 3, -1]‘

请实现此功能的函数。

```
def replace_with_max_right(arr):  
    max_right = -1  
    # 从后向前遍历  
    for i in range(len(arr) - 1, -1, -1):  
        current = arr[i]  
        arr[i] = max_right  
        if current > max_right:  
            max_right = current  
    return arr
```

- 时间复杂度 **: $O(n)$, 其中 n 是数组的长度, 因为只需要一次遍历。
- ** 空间复杂度 **: $O(1)$, 只使用了常数空间。