

期末模拟试卷-1

一、 单选题 （共 15 题， 30 分）

1、 以下表达式 $3 + 2 * 4 ** 2 // 3$ 的计算结果是多少？

(2.0)

A、 13

B、 19

C、 21

D、 35

正确答案： A

解析：

2、 下列表达式求值后， 结果的类型是什么？

```
res = True + 2
```

(2.0)

A、 bool

B、 int

C、 float

D、 str

正确答案： B

解析：

3、 在 Python 中， 当使用逻辑与运算符 and 时， 下列表达式的求值结果是多少？

```
x = 0
```

```
result = (x != 0) and (10 / x)
```

(2.0)

- A、 抛出 ZeroDivisionError 异常
- B、 True
- C、 False
- D、 None

正确答案： C

解析：

4、 以下哪个选项是合法的 Python 变量名？

(2.0)

- A、 my_var
- B、 2name
- C、 my-var
- D、 class

正确答案： A

解析：

5、 以下表达式的求值结果是什么？

```
expr = int('1011', 2)
```

(2.0)

- A、 11
- B、 1011
- C、 3
- D、 抛出 ValueError 异常

正确答案： A

解析：

6、 以下表达式求值结果是多少？

```
result = sum(range(1,5))
```

(2.0)

- A、 10

- B、 9
- C、 15
- D、 抛出 `TypeError` 异常

正确答案： A

解析：

7、 以下关于 `input()` 函数的描述中， 哪一项是正确的？

(2.0)

- A、 `input()` 函数返回用户输入的字符串类型值
- B、 `input()` 函数返回用户输入的整数类型值
- C、 `input()` 函数会返回 `None`
- D、 `input()` 函数需要在参数中指定类型， 否则报错

正确答案： A

解析：

8、 以下代码执行后 `lst` 的值是什么？

```
lst = [1, 2, 3, 4, 5]
```

```
lst[1:3] = [10, 20, 30]
```

(2.0)

- A、 `[1, 10, 20, 30, 4, 5]`
- B、 `[1, 10, 20, 30, 5]`
- C、 `[1, 2, 10, 20, 30, 4, 5]`
- D、 `[1, 10, 20, 3, 4, 5]`

正确答案： A

解析：

9、 以下代码执行后 `lst2` 的值是什么？

```
lst1 = [1, 2, 3]
```

```
lst2 = lst1
```

```
lst1.append(4)
```

(2.0)

- A、 [1, 2, 3]
- B、 [1, 2, 3, 4]
- C、 []
- D、 抛出 NameError 异常

正确答案： B

解析：

10、 lst = [3, 1, 2]

lst.sort(reverse=True)

执行后 lst 的值是什么？

(2.0)

- A、 [3, 2, 1]
- B、 [1, 2, 3]
- C、 [3, 1, 2]不变
- D、 抛出 TypeError 异常

正确答案： A

解析：

11、 t = (1, 2, 3)

尝试修改元组元素

t[1] = 20

如果将注释行解除注释，那么运行结果会如何？

(2.0)

- A、 t 变为(1,20,3)
- B、 t 变为(1,2,3,20)
- C、 不会有任何影响
- D、

抛出 TypeError 异常

正确答案： D

解析：

12、以下代码执行后 s 的值是什么？

```
s = {10, 20, 30}
```

```
s.discard(40)
```

(2.0)

A、 {10, 20, 30, 40}

B、 {10, 20, 30} (不变)

C、 抛出 KeyError 异常

D、 {10, 20}

正确答案： B

解析：

13、以下代码执行后会产生什么结果？

```
d = {}
```

```
d[[1,2]] = 3
```

(2.0)

A、 d = {[1,2]:3}

B、 d = {} (不变)

C、 TypeError 异常

D、 KeyError 异常

正确答案： C

解析：

14、from functools import reduce

```
nums = [2, 3]

res = reduce(lambda a,b: a*b, nums, 10)
```

执行后 res 的值是多少？

(2.0)

A、 60

B、 6

C、 10

D、 2

正确答案： A

解析：

15、try:

```
    int("abc")
```

```
except ValueError:
```

```
    print("ValueError 被捕获")
```

```
except Exception:
```

```
    print("Exception 被捕获")
```

执行后会打印什么？

(2.0)

A、 "Exception 被捕获"

B、 "ValueError 被捕获"

C、 不打印任何东西

D、 抛出未捕获异常

正确答案： B

解析：

二、 填空题 （共 5 题， 20 分）

16、代码如下，执行结果如何？

```
val = 0

if val:
    res1 = 'True branch'
else:
    res1 = 'False branch'

if len(res1) > 10:
    res2 = len(res1)
else:
    res2 = 100 - len(res1)

print(res1)
print(res2)
```

(4.0)

正确答案:

第 1 空:

False branch

第 2 空:

12

解析:

17、代码如下，请问执行后打印的两个结果分别是什么？

```
numbers = [1,2,3,4,5]

acc = 0

mul = 1

for n in numbers[1:4]:

    acc += n

for n in numbers[:2]:

    mul *= n

print(acc)

print(mul)
```

(4.0)

正确答案:

第 1 空:

9

第 2 空:

2

解析:

18、代码如下，请问执行后打印的两个结果分别是什么？

```
num = 3

product = 1

while num > 0:
```



```
    product *= num

    num -= 1

print(product)

print(num)
```

(4.0)

正确答案:

第 1 空:

6

第 2 空:

0

解析:

19、代码如下，请问执行后打印的两个结果分别是什么？

```
numbers = [10, 20, 30, 40]

res = 0

idx = 0

for num in numbers:

    idx += 1

    if num > 25:

        break

    res += num

print(res)
```

```
print(idx)
```

(4.0)

正确答案:

第 1 空:

30

第 2 空:

3

解析:

20、代码如下，请问执行后打印的两个结果分别是什么？

```
import copy
lst = [10, [20,30], 40]
lst_deep = copy.deepcopy(lst)
lst_deep[1].append(99)
print(lst[1])
print(lst_deep[1])
```

(4.0)

正确答案:

第 1 空:

[20, 30]

第 2 空:

[20, 30, 99]

解析：

三、 判断题 （共 10 题， 20 分）

21、将多个相关的模块放入同一目录并添加__init__.py 文件后，即可形成一个包（package），这样可实现更加清晰的代码组织。

（2.0）

正确答案： 正确

解析：

22、TypeError 在不支持的运算类型之间进行操作时会引发，例如试图将整数与字符串直接相加。

（2.0）

正确答案： 正确

解析：

23、requirements.txt 文件用于记录项目所需的第三方包及其具体版本，以便在不同环境中复现相同的依赖关系。

（2.0）

正确答案： 正确

解析：

24、使用'a'模式打开文件时，如果文件不存在，Python 会自动创建该文件。

（2.0）

正确答案： 正确

解析：

25、在 Python 中，所有关键字参数（keyword arguments）必须在位置参数（positional arguments）之前传递给函数。

（2.0）

正确答案： 错误

解析：

26、在 Python 中，绝对路径始终以根目录（如 Windows 中的驱动器字母或 Unix/Linux 中的斜杠）开头，而相对路径则不以根目录开头。

（2.0）

正确答案： 正确

解析：

27、在 Python 中，全局变量可以在任何函数内部被直接修改，而无需使用 global 关键字。

（2.0）

正确答案： 错误

解析：

28、exec 函数可以执行字符串形式的 Python 代码，并且可以返回执行结果。

（2.0）

正确答案： 错误

解析：

29、在使用 str.format() 方法时，双花括号{{和}}可以用于在最终字符串中表示单个花括号{和}。

（2.0）

正确答案： 正确

解析：

30、在 Python 中，浮点数的运算顺序不会影响最终的结果，因为所有浮点运算都是精确的。

（2.0）

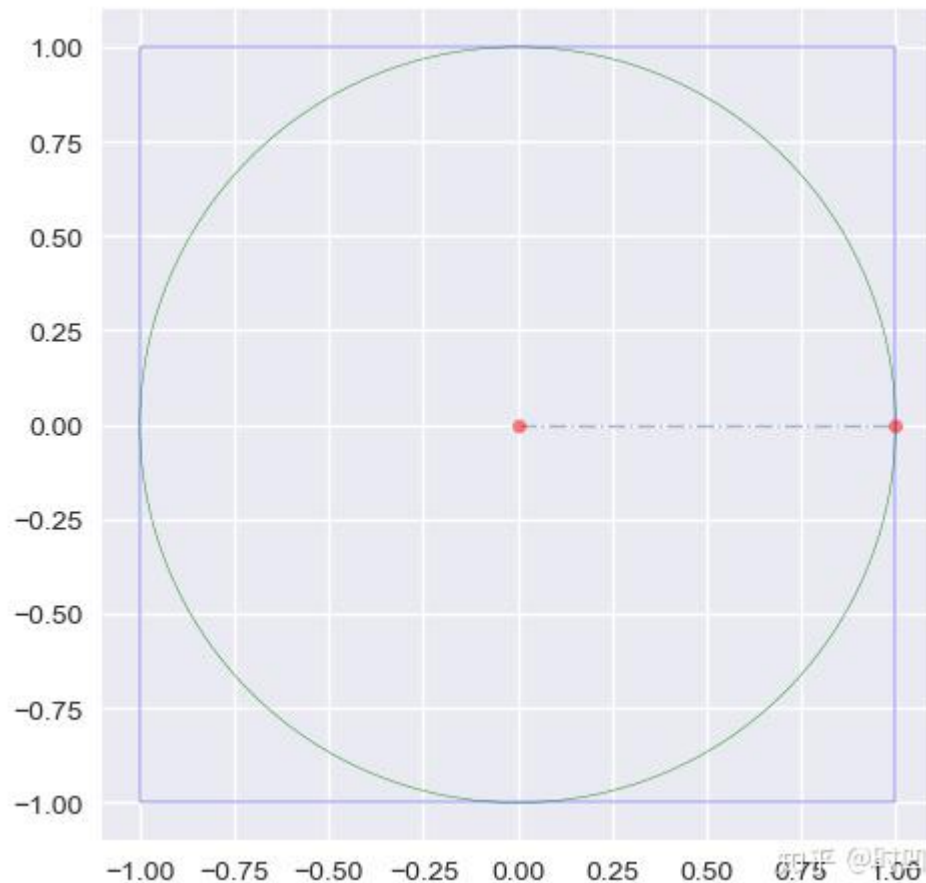
正确答案： 错误

解析：

四、 简答题 （共 3 题，30 分）

31、蒙特卡罗法估算圆周率（10 分）

以原点为圆心的单位圆内接于一个边长为 2 的正方形。请编写程序，使用蒙特卡罗随机采样方法估算圆周率 π 的值。具体要求如下：



1. 从正方形区域内随机采样 100 万个点，计算落在圆内的点的比例，进而估算 π 值；（4 分）
2. 将程序运行结果与 `math.pi` 进行对比，计算并输出估算值与真实值的相对误差；（3 分）
3. 使用合适的格式化方法输出结果，真实值和估算值均保留 8 位小数，相对误差保留 4 位小数。（3 分）

（10.0）

正确答案：

```
import math
import random
```

```

def estimate_pi(num_points):
    # 计算π的估计值（4分）
    hit = 0 # 落在圆内的点数
    for _ in range(num_points):
        # 在[-1,1]×[-1,1]的正方形区域内随机采样
        x = random.uniform(-1.0, 1.0)
        y = random.uniform(-1.0, 1.0)
        # 判断点是否落在单位圆内
        if x**2 + y**2 <= 1:
            hit += 1

    pi_estimate = 4 * (hit / num_points)

    # 计算相对误差（3分）
    relative_error = abs(pi_estimate - math.pi) / math.pi * 100

    # 格式化输出结果（3分）
    print(f"实际的π值: {math.pi:30.8f}")
    print(f"估算的π值: {pi_estimate:30.8f}")
    print(f"相对误差: {relative_error:30.4f}%")

    return pi_estimate, relative_error

# 运行程序进行验证
num_points = 1_000_000
pi_est, error = estimate_pi(num_points)

```

解析：

[\[蒙特卡罗题目解析.ipynb\]](#)

32、个人所得税计算器（10 分）

编写一个 Python 程序，计算个人年度所得税。具体要求如下：

1. 设计函数接收年收入金额（整数，单位：元）（2 分）

2. 根据以下分级税率表计算应缴税额（5 分）：

不超过 36000 元的部分：3%

超过 36000 元至 144000 元的部分：10%

超过 144000 元至 300000 元的部分：20%

超过 300000 元至 420000 元的部分：25%

超过 420000 元至 660000 元的部分：30%

超过 660000 元至 960000 元的部分：35%

超过 960000 元的部分：45%

3. 输出详细的税额计算过程和最终应缴税额，结果保留 2 位小数（3 分）

(10.0)

正确答案：

```
def calculate_tax(annual_income):  
    """  
    计算个人所得税  
    参数: annual_income - 年收入金额（元）  
    返回: 税额明细字典和总税额  
    """  
  
    # 定义税率表: (起始金额, 结束金额, 税率)  
    tax_rates = [  
        (0, 36000, 0.03),  
        (36000, 144000, 0.10),  
        (144000, 300000, 0.20),  
        (300000, 420000, 0.25),  
        (420000, 660000, 0.30),  
        (660000, 960000, 0.35),  
        (960000, float('inf'), 0.45)  
    ]  
  
    total_tax = 0  
    tax_details = []  
  
    # 逐级计算税额  
    remaining_income = annual_income  
    for start, end, rate in tax_rates:
```

```

        if remaining_income <= 0:
            break

        taxable_amount = min(remaining_income, end - start)
        if taxable_amount > 0:
            tax = taxable_amount * rate
            total_tax += tax
            tax_details.append({
                'range': f"{start:,}~{end if end != float('inf') else '以上'}",
                'taxable_amount': taxable_amount,
                'rate': rate,
                'tax': tax
            })

        remaining_income -= taxable_amount

    return tax_details, total_tax

def print_tax_report(annual_income):
    """打印税收计算报告"""
    print("\n===== 个人所得税计算报告 =====")
    print(f"年收入: {annual_income:,.2f}元")
    print(f"\n 分级计算过程: ")

    tax_details, total_tax = calculate_tax(annual_income)

    for detail in tax_details:
        print(f"应税区间: {detail['range']}")
        print(f"应纳税金额: {detail['taxable_amount']:,.2f}元")
        print(f"适用税率: {detail['rate']:.1%}")
        print(f"应缴税额: {detail['tax']:,.2f}元")
        print("-" * 30)

    print(f"\n 总应缴税额: {total_tax:,.2f}元")
    print(f"税后收入: {annual_income - total_tax:,.2f}元")
    print("=" * 30)

# 测试代码
test_incomes = [30000, 100000, 200000, 500000, 1000000]
for income in test_incomes:
    print_tax_report(income)

```

解析:

[\[个税计算题目解析.ipynb\]](#)

33、销售数据汇总（10 分）

在商业数据分析中，汇总销售数据是常见的任务。假设有一个名为 `sales_data.txt` 的文本文件，文件中每一行记录一个销售订单的信息，格式如下：

订单编号,客户名称,销售金额

例如：

001,客户 A,5000

002,客户 B,7000

003,客户 A,3000

请编写一个程序，读取 `sales_data.txt` 文件，计算每个客户的总销售金额，并将结果写入一个名为 `customer_sales_summary.txt` 的文件中。输出文件的格式应为：

客户名称：总销售金额

例如：

客户 A：8000

客户 B：7000

评分点：

- 正确读取并解析文本文件内容（3 分）
- 正确计算每个客户的总销售金额（4 分）
- 正确将结果写入输出文件，格式符合要求（3 分）

（10.0）

正确答案：

```
def summarize_sales(input_file, output_file):
    customer_sales = {}
    with open(input_file, 'r', encoding='utf-8') as f:
        for line in f:
            line = line.strip()
            if not line:
                continue # 跳过空行
            parts = line.split(',')
            if len(parts) != 3:
                continue # 跳过格式错误的行
            order_id, customer, amount = parts
            try:
                amount = float(amount)
            except ValueError:
                continue # 跳过金额无法转换的行
            if customer in customer_sales:
                customer_sales[customer] += amount
            else:
                customer_sales[customer] = amount
    with open(output_file, 'w', encoding='utf-8') as f:
        for customer, total in customer_sales.items():
            f.write(f"{customer}: {int(total)}\n")

# 示例运行
summarize_sales('sales_data.txt', 'customer_sales_summary.txt')
```

解析：

[\[销售数据汇总题目解析.ipynb\]](#)