

2024 秋招 Python 试题答案 (A)

一、选择题（5 分/题，共 10 题，50 分）

1. Python 不支持的数据类型有 (A)。

- A. char
- B. bool
- C. complex
- D. list

2. 关于 Python 的浮点数类型，以下选项中描述错误的是 (C)。

- A. 浮点数类型与数学中实数的概念一致，表示带有小数的数值
- B. sys.float_info 可以详细列出 Python 解释器所运行系统的浮点数各项参数
- C. Python 语言的浮点数可以不带小数部分
- D. 浮点数有两种表示方法：十进制表示和科学计数法

3. 下列是下载包的正确语法的是 (B)。

- A. install numpy
- B. pip install numpy
- C. install pip numpy
- D. install conda numpy

4. 在 Python 中，输出小数并保留小数点后四位的操作是 (D)。

- A. 使用 print(%f%float)
- B. 使用 print(f' {float} ')
- C. 使用 print(f' {:.4f}' .format(float))
- D. 使用 print(' {:.4f}' .format(float))

5. 以下不能创建一个字典的语句是 (C)。

- A. dict1 = {}
- B. dict2 = { 3 : 5 }

- C. dict3 = {[1, 2, 3]: “ajabj” }
- D. dict4 = {(1, 2, 3): “ajabj” }

6. 对于序列 s，能够返回序列 s 中第 i 到 j 以 k 为步长的元素子序列的表达是 (D)。

- A. s[i, j, k]
- B. s(i, j, k)
- C. s[i; j; k]
- D. s[i:j:k]

7. 在 Python 中，程序运行结果为 (B)。

```
a = 100  
b = 14  
print(divmod(a, b))
```

- A、(7, 0)
- B、(7, 2)
- C、[7, 2]
- D、None

8. 在 Python 中，程序运行结果为 (D)。

```
list1 = {'1':1, '2':2}  
list2 = list1
```

```
list1['1'] = 5  
sum = list1['1'] + list2['1']  
print(sum)
```

- A、1
- B、2
- C、7
- D、10

9. 在 Python 中有

```
s=['a', 'b']
```

```
s.append([1, 2])  
s.insert(1, 7); 执行以上代码后, s 值为 (D)。
```

- A. ['a', 7, 'b', 1, 2]
- B. [[1, 2], 7, 'a', 'b']
- C. [1, 2, 'a', 7, 'b']
- D. ['a', 7, 'b', [1, 2]]

10. 给出如下代码

```
import random as ran  
  
listV = []  
  
ran.seed(100)  
  
for i in range(10):  
    i = ran.randint(100, 999)  
  
listV.append(i)
```

以下选项中能输出随机列表元素最大值的是 (C)。

- A
print(listV.max())
- B
print(listV.pop(i))
- C
print(max(listV))
- D
print(listV.reverse(i))

选择题答案: ACBDCDBDDC

二、编程题 (共 4 题, 50 分)

1. (10 分) 使用给定的整数 n, 编写一个程序生成一个包含($i, i*i$)的字典, 该字典包含 1 到 n 之间的整数(两者都包含)。然后程序应该打印字典。

假设向程序提供以下输入:8

则输出为：

```
{1:1, 2:4, 3:9, 4:16, 5:25, 6:36, ,7:49, 8:64}  
print('请输入一个数字：')  
n=int(input())  
d=dict()  
for i in range(1,n+1):  
    d[i]=i*i  
  
print(d)
```

循环写对，得 7 分

有小错误或者写出完整思路，得 2 分

动笔，得 1 分

分数累加

2. (10 分) 编写一个程序，接受一系列逗号分隔的 4 位二进制数作为输入，然后检查它们是否可被 5 整除。可被 5 整除的数字将以逗号分隔的顺序打印。

例：

0100, 0011, 1010, 1001

那么输出应该是：

1010

注意：假设数据由控制台输入。

```
value = []  
print('请输入逗号分隔的 4 位二进制数：')  
items=[x for x in input().split(',')]  
for p in items:  
    intp = int(p, 2)  
    # print(intp)  
    if not intp%5:  
        value.append(p)  
print (','.join(value))
```

转化二进制到十进制，得 4 分

判断是否被五整除，得 3 分

有小错误或者写出完整思路，得 2 分

动笔，得 1 分

3. 如果已知英制长度的英尺 foot 和英寸 inch 的值，那么对应的米是 $(foot+inch/12) \times 0.3048$ 。现在，如果用户输入的是厘米数，那么对应英制长度的英尺和英寸是多少呢？别忘了 1 英尺等于 12 英寸。

输入格式：

输入在一行中给出 1 个正整数，单位是厘米。

输出格式：

在一行中输出这个厘米数对应英制长度的英尺和英寸的整数值，中间用空格分开。英寸的值应小于 12。

```
# 读取输入
cm = int(input())
# 将厘米转换为米
meters = cm / 100
# 将米转换为英尺
feet = meters / 0.3048
# 计算英尺的整数部分
feet_int = int(feet)

# 计算剩余的英寸
remaining_inches = (feet - feet_int) * 12

# 确保英寸数小于 12
while remaining_inches >= 12:
    feet_int += 1
    remaining_inches -= 12

# 转换并输出结果
feet, inches = feet_int, int(remaining_inches)
print(f"{feet} {inches}")
```

厘米转化成米，得 1 分

米转化成英尺，得 2 分

计算英寸，得 2 分

确保英寸数，得 2 分

有小错误或者写出完整思路，得 2 分

动笔，得 1 分

4. 乌龟与兔子进行赛跑，跑场是一个矩型跑道，跑道边可以随地进行休息。乌龟每分钟可以前进 3 米，兔子每分钟前进 9 米；兔子嫌乌龟跑得慢，觉得肯定能跑赢乌龟，于是，每跑 10 分钟回头看一下乌龟，若发现自己超过乌龟，就在路边休息，每次休息 30 分钟，否则继续跑 10 分钟；而乌龟非常努力，一直跑，不休息。假定乌龟与兔子在同一起点同一时刻开始起跑，请问 T 分钟后乌龟和兔子谁跑得快？

输入格式：

输入在一行中给出比赛时间 T（分钟）。

输出格式：

在一行中输出比赛的结果：乌龟赢输出@_@，兔子赢输出^_^，平局则输出-_-；后跟 1 空格，再输出胜利者跑完的距离。

```
def race(T):
    # 初始化位置
    turtle_pos = 0
    rabbit_pos = 0
```

```
# 初始化兔子的休息时间
rabbit_rest = 0
for t in range(1, T + 1):
    # 乌龟每分钟前进 3 米
    turtle_pos += 3
    # 兔子每分钟前进 9 米，但每 10 分钟可能休息 30 分钟
    if rabbit_rest > 0:
        rabbit_rest -= 1
    else:
        rabbit_pos += 9
    # 每跑 10 分钟，兔子休息的条件
    if t % 10 == 0 and rabbit_pos > turtle_pos:
        rabbit_rest = 30
# 比较结果
if turtle_pos > rabbit_pos:
    return "@_@"
elif turtle_pos < rabbit_pos:
    return "^_^"
else:
    return "-_-"
# 读取输入
T = int(input())
# 输出结果
print(race(T))
```

写出乌龟在 T 时间的位置，得 3 分
写出图纸在 T 时间的位置，得 3 分
输入格式正确，得 1 分
有小错误或者写出完整思路，得 2 分
动笔，得 1 分