

# 2024 秋招 Python 试题答案 (B)

## 一、选择题 (5 分/题, 共 10 题, 50 分)

1. Python 支持的变量名有 (C)。

- A. def
- B. 1day
- C. \_st
- D. last time

2. 以下选项中, 不是 Python 数据类型的是 (A)。

- A. 实数
- B. 列表
- C. 整数
- D. 字符串

3. 关于 Python 的复数类型, 以下选项中描述错误的是 (B)。

- A. 复数类型表示数学中的复数
- B. 对于复数 `z`, 可以用 `z.imag` 获得实数部分
- C. 复数的虚数部分通过后缀“J”或“j”来表示
- D. 对于复数 `z`, 可以用 `z.real` 获得实数部分

4. 在 Python 中, 不同的数据, 需要定义不同的数据类型, 可用方括号 `[]` 来定义的是 (A)。

- A. 列表
- B. 元组
- C. 集合
- D. 字典

5. 字符串 `lis='goodme'`, 其切片 `lis[-1:1:-1]` 结果为 (C)。

- A. emdoog
- B. emdoo

- C. emdo
- D. goodme

6. 设序列 `s`，以下选项中对 `max(s)` 的描述正确的是（ C ）。

- A. 返回序列 `s` 的最大元素，如果有多个相同，则返回一个元组类型
- B. 返回序列 `s` 的最大元素，如果有多个相同，则返回一个列表类型
- C. 返回序列 `s` 的最大元素，但要求 `s` 中元素之间可比较
- D. 一定能够返回序列 `s` 的最大元素

7. 元组变量 `t=("cat", "dog", "tiger", "human")`，`t[::-1]` 的结果是（ D ）。

- A.
- B. `['human', 'tiger', 'dog', 'cat']`
- C. 运行出错
- D. `('human', 'tiger', 'dog', 'cat')`

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 Python3 中。下列程序运行结果说明正确的是（ D ）。

```
strs = 'abcd12efg'
print(strs.upper().title())
```

- A. 'ABCD12EFG'

- B. 'Abc12efg'
- C. 语法错误
- D. 'Abcd12Efg'

10. 执行以下程序，当用户输入 0 时，输出结果为 (D)。

```
dividend = 1
divide = int(input())
try:
    result = dividend / divide
    print(1, end=" ")
except ZeroDivisionError:
    print(2, end=" ")
except Exception:
    print(3, end=" ")
else:
    print(4)
```

- A. 1 2
- B. 2 4
- C. 2 3
- D. 2

选择题答案：CABACCDDDD

## 二、编程题（共 4 题，50 分）

温馨提示：编程题不会写可以将思路写下来哦，思路正确也有分

编程题是按照步骤给分，各位同学不要吝啬自己的笔墨哦

1. (10 分) 给你一个非负整数  $x$ ，计算并返回  $x$  的算术平方根。由于返回类型是整数，结果只保留整数部分，小数部分将被舍去。

```
import math
x = int(input('请输入数字'))
if x < 0:
    print('数字错误')
else:
    result = math.floor(math.sqrt(x))
print(f'{result} 是 {x} 的算术平方根的整数部分')
```

计算出算术平方根，得 5 分  
判断数字是否大于 0，得 2 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，计算  $a + aa + aaa + aaaa$  的值，给定的数字作为  $a$  的值。假设程序提供了以下输入：9

然后，输出应该是：11106

提示：如果输入数据被提供给问题，则应该假定它是控制台输入

```
print('请输入一个数字：')
a = input()
n1 = int( "%s" % a )
n2 = int( "%s%s" % (a,a) )
n3 = int( "%s%s%s" % (a,a,a) )
n4 = int( "%s%s%s%s" % (a,a,a,a) )
print (n1+n2+n3+n4)
```

计算出数值，得 7 分  
有小错误或者写出完整思路，得 2 分  
动笔，得 1 分

3. （15 分）一条蠕虫长 1 寸，在一口深为  $N$  寸的井的底部。已知蠕虫每 1 分钟可以向上爬  $U$  寸，但必须休息 1 分钟才能接着往上爬。在休息的过程中，蠕虫又下滑了  $D$  寸。就这样，上爬和下滑重复进行。请问，蠕虫需要多长时间才能爬出井？

这里要求不足 1 分钟按 1 分钟计，并且假定只要在某次上爬过程中蠕虫的头部到达了井的顶部，那么蠕虫就完成任务了。初始时，蠕虫是趴在井底的（即高度为 0）。

输入格式：

输入在一行中顺序给出 3 个正整数  $N$ 、 $U$ 、 $D$ ，其中  $D < U$ ， $N$  不超过 100。

输出格式：

在一行中输出蠕虫爬出井的时间，以分钟为单位。

```
# 读取输入
N, U, D = map(int, input().split())
current_height = 0
total_time = 0
```

```

while current_height < N:
    # 蠕虫向上爬 U 寸
    current_height += U
    total_time += 1
    # 检查蠕虫是否已经爬出井
    if current_height >= N:
        break
    # 蠕虫休息 1 分钟，下滑 D 寸
    current_height -= D
    total_time += 1
# 输出结果
print(total_time)

```

逻辑正确，得 11 分

有小错误或者写出完整思路，得 3 分

动笔，得 1 分

4. (15 分) 输入一个字符串和一个非负整数 N，要求将字符串循环左移 N 次。

**输入格式：**

输入在第 1 行中给出一个不超过 100 个字符长度的、以回车结束的非空字符串；第 2 行给出非负整数 N。

**输出格式：**

在一行中输出循环左移 N 次后的字符串。

```

# 读取输入
s = input().strip()
n = int(input())
# 计算实际需要移动的位数
n = n % len(s)
# 将字符串转换为列表
s_list = list(s)
# 进行左移操作
s_list = s_list[n:] + s_list[:n]
# 将列表转换回字符串,输出结果
print("".join(s_list))

```

逻辑正确，得 11 分（如果写的循环左移，得 7 分）

有小错误或者写出完整思路，得 3 分

动笔，得 1 分