



Python 四级

2024 年 12 月

1 单选题（每题 2 分，共 30 分）

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	C	A	D	C	A	C	D	C	B	A	B	A	D	B	A

第 1 题 2024年10月8日，诺贝尔物理学奖“意外地”颁给了两位计算机科学家约翰·霍普菲尔德（John J. Hopfield）和杰弗里·辛顿（Geoffrey E. Hinton）。这两位科学家的主要研究方向是（ ）。

- A. 天体物理
- B. 流体力学
- C. 人工智能
- D. 量子理论

第 2 题 计算机系统中存储的基本单位用B来表示，比如某个照片大小为3MB，其中B代表的是（ ）。

- A. Byte
- B. Block
- C. Bulk
- D. Bit

第 3 题 执行下面Python代码后，输出的结果是？（ ）

```
1 items = [('item1', 2), ('item2', 3), ('item3', 1)]
2 sorted_items = sorted(items, key=lambda x: (-x[1], x[0]))
3 print(sorted_items)
```

- A. [('item3', 1), ('item1', 2), ('item2', 3)]
- B. [('item1', 2), ('item2', 3), ('item3', 1)]
- C. [('item2', 3), ('item3', 1), ('item1', 2)]
- D. [('item2', 3), ('item1', 2), ('item3', 1)]

第 4 题 以下Python代码实现的排序算法的时间复杂度是？（ ）

```

1 def func_sort(arr):
2     for i in range(1, len(arr)):
3         key = arr[i]
4         j = i-1
5         while j >= 0 and key < arr[j]:
6             arr[j+1] = arr[j]
7             j -= 1
8         arr[j+1] = key

```

- A. $O(n)$
- B. $O(2n)$
- C. $O(n^2)$
- D. $O(n^3)$

第 5 题 执行下面Python代码后，输出的结果是？（ ）

```

1 z = 5
2
3
4 def outer():
5     def inner():
6         global z
7         z = 15
8
9     inner()
10    print(z, end="#")
11
12
13 outer()
14 print(z, end="#")

```

- A. 15#15#
- B. 15#5#
- C. 5#15#
- D. 5#5#

第 6 题 执行下面Python代码后，输出的结果是？（ ）

```

1 def double_elements(lst):
2     for i in range(len(lst)):
3         lst[i] *= 2
4     return lst
5
6
7 lstB = [1, 2, 3]
8 double_elements(lstB)
9 print(lstB, double_elements(lstB))

```

- A. [1, 2, 3] [2, 4, 6]

- B. [2, 4, 6] [4, 8, 12]
- C. [4, 8, 12] [4, 8, 12]
- D. [1, 2, 3] [4, 8, 12]

第7题 执行下面Python代码后，会发生什么？（）

```
1 my_list = [1, 2, 3]
2 print(my_list[3])
```

- A. 代码正常执行，输出3
- B. 抛出TypeError异常
- C. 抛出ValueError异常
- D. 抛出IndexError异常

第8题 执行下面Python代码后，输出的结果是？（）

```
1 try:
2     my_dict = {'a': 1, 'b': 2}
3     print(my_dict['c'])
4 except KeyError:
5     print("10", end="#")
6 else:
7     print("20", end="#")
8 finally:
9     print("30", end="#")
```

- A. 10#
- B. 20#
- C. 10#30#
- D. 20#30#

第9题 执行下面Python代码后，输出的结果可能是？（）

```
1 my_dic = {'a': [1, 2, 3], 'b': (4, 5, 6)}
2 my_dic['a'][0] = 10
3 my_dic['b'] += (7,)
4 print(my_dic)
```

- A. {'a': [1, 2, 3], 'b': (4, 5, 6, 7)}
- B. {'a': [10, 2, 3], 'b': (4, 5, 6, 7)}
- C. {'a': [10, 2, 3], 'b': (4, 5, 6)}
- D. {'a': [10, 2, 3], 'b': (4, 5, 6, 7, 7)}

第10题 执行下面Python代码后，输出的结果是？（）

```

1 def func(a, *, b):
2     print("{},{}".format(a, b))
3
4
5 func(1, b=2)

```

- A. 1,2
- B. 2,1
- C. 抛出异常
- D. None

第 11 题 执行下面Python代码后，输出的结果是？（）

```

1 def func(**kwargs):
2     return {"name": kwargs.get("name", "Guest"),
3           "age": kwargs.get("age", 18)}
4
5
6 user = func(name="Bob")
7 print(user["name"])

```

- A. Guest
- B. Bob
- C. 18
- D. None

第 12 题 在一个棋盘上，有一堆麦子。根据棋盘的规则，第1格放1粒麦子，第2格放2粒麦子，第3格放4粒麦子，以此类推，每增加一格，麦子的数量翻倍。下列程序用来计算在第 $n(n < 64)$ 格有多少粒麦子，其中横线处填写的代码为？（）

```

1 lst = [0] * 64
2 lst[1] = 1
3
4
5 def calculate_wheat(n):
6     if n <= 1:
7         return lst[n]
8     i = 2
9     while i <= n:
10        lst[i] = _____ # 填写代码
11        i += 1
12    return lst[i - 1]

```

- A. $lst[i - 1] * 2$
- B. $lst[i - 2] * 2$
- C. $lst[i - 1] + lst[i - 2]$
- D. $lst[-1] * 2$

第 13 题 小杨相信数字“1”、“3”、“5”能给他带来好运，不喜欢数字“7”。他设计了一个计算数字幸运值的程序。对于数字中的所有字符，字符“1”幸运值为 5，字符“3”幸运值为 6，字符“5”幸运值为 7，字符“7”的幸运值为-3，其他字符的幸运值为 0，数字的幸运值即为其所有字符的幸运值之和。请在下面程序空白处填上正确内容以实现计算某数字幸运值的功能。

```

1  dic = {"1": 5, "3": 6, "5": 7, "7": -3}
2  num = input("请输入数字: ")
3  ans = 0
4  for i in range(len(num)):
5      ch = num[i]
6      if _____ : # 填写代码 ①
7          _____ # 填写代码 ②
8  print("该数字的幸运值为: ", ans)

```

- A. ① `ch in dic.values()` ② `ans += dic[ch]`
- B. ① `ch in dic.items()` ② `ans += dic[ch]`
- C. ① `ch in dic.keys()` ② `ans = dic[ch]`
- D. ① `ch in dic` ② `ans += dic[ch]`

第 14 题 打开一个由小写英文和数字组成的文本文件notes.txt，将该文件中的每一个字母加密后写入到一个新文件encrypted_notes.txt，加密的方法是：a变成b，b变成c，……，z变成a，其它字符不变化。请在下面程序空白处填上正确内容以实现程序功能。

```

1  file = open("notes.txt", "r")
2  line = file.readline()
3  list = []
4  while line:
5      for i in range(len(line)):
6          if line[i].islower():
7              jm = _____ # 填写代码 ①
8              list.append(chr(jm))
9          else:
10             list.append(line[i])
11         line = file.readline()
12     s = _____ # 填写代码 ②
13     file = open("encrypted_notes.txt", "w+")
14     file.write(s)
15     file.close()

```

- A. ① `(ord(line[i]) + 1) % 26` ② `''.join(list)`
- B. ① `ord('a') + (ord(line[i]) + 1 - ord('a')) % 26` ② `''.join(list)`
- C. ① `(ord(line[i]) + 1) % 26` ② `str(list)`
- D. ① `ord('a') + (ord(line[i]) + 1 - ord('a')) % 26` ② `str(list)`

第 15 题 假设你管理一个实验室，需要记录物品的领用情况。每次领用时，你会记录下领用的日期、物品名称以及领用人的姓名。这个信息将被保存在一个名为lab_issuance.txt的文本文件中。其要求如下：①在文件末尾追加新的物品领用记录；②可以查看当前所有的物品领用记录。请在下面程序空白处填上正确内容以实现程序功能。

```

1  def append_record(date, item, name):
2      with open("lab_issuance.txt", _____) as file: # 填写代码 ①

```

```

3     file.write(f"{date},{item},{name}\n")
4
5 def read_records():
6     with open("lab_issuance.txt", "r") as file:
7         records = _____ # 填写代码 ②
8         for record in records:
9             print(record, end="")
10
11 # 示例操作
12 append_record("2024-11-05", "Computer", "Tom")
13 append_record("2024-11-06", "Pen", "Jack")
14 read_records()

```

- A. ① "a" ② file.readlines()
- B. ① "a" ② file.readline()
- C. ① "w" ② file.readlines()
- D. ① "w" ② file.readline()

2 判断题（每题 2 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	√	×	√	×	√	√	×	√	×	√

第 1 题 在Windows的资源管理器中为已有文件A建立副本的操作是Ctrl+C，然后Ctrl+V。

第 2 题 执行下面Python代码后，输出的结果是14。

```

1 a = 5
2 b = 3
3 result = (a & b) << 1
4 print(result)

```

第 3 题 执行下面Python代码，调用函数get_max可以得到一个None类型的数据。

```

1 def get_max(*args):
2     if not args:
3         return None
4     return max(args)
5
6
7 print(type(get_max()))

```

第 4 题 执行下面Python代码后，输出的结果为[1, 3, 5, 7]。

```

1 print(list(filter(lambda x: x % 2 == 0, [1, 2, 3, 4, 5, 6, 7, 8])))

```

第 5 题 下面这段程序的时间复杂度为常数阶 $O(1)$ 。

```

1 def fun(n):
2     print("Hello, World!")

```

第 6 题 如果在函数内部对一个变量进行赋值操作，那么该变量默认为局部变量。

第7题 在Python中，已执行 `list_of_tuples = [(1, 2), (3, 4)]`，如果执行 `list_of_tuples[0][1] = 5` 将不会报错。

第8题 在Python中，`[(i, i ** 2) for i in range(5)] + [7,]` 是合法的表达式。（）

第9题 根据下面Python函数定义，调用func()函数时如果参数分别为str和tuple类型，会报错。

```
1 def func(a, b):  
2     return a + str(b)
```

第10题 在Python中，使用 `with open(file, 'w') as f:` 语句打开文件后，即使发生异常，文件资源通常也会被正确关闭。

3 编程题（每题 25 分，共 50 分）

3.1 编程题 1

- 试题名称: Recamán
- 时间限制: 2.0 s
- 内存限制: 512.0 MB

3.1.1 题目描述

小杨最近发现了有趣的 Recamán 数列，这个数列是这样生成的：

- 数列的第一项 a_1 是 1；
- 如果 $a_{k-1} - k$ 是正整数并且没有在数列中出现过，那么数列的第 k 项 a_k 为 $a_{k-1} - k$ ，否则为 $a_{k-1} + k$ 。

小杨想知道 Recamán 数列的前 n 项从小到大排序后的结果。手动计算非常困难，小杨希望你能帮他解决这个问题。

3.1.2 输入格式

第一行，一个正整数 n 。

3.1.3 输出格式

一行， n 个空格分隔的整数，表示 Recamán 数列的前 n 项从小到大排序后的结果。

3.1.4 样例

3.1.5 输入样例 1

```
1 | 5
```

3.1.6 输出样例 1

```
1 | 1 2 3 6 7
```

3.1.7 输入样例 2

```
1 | 8
```

3.1.8 输出样例 2

```
1 | 1 2 3 6 7 12 13 20
```

3.1.9 样例解释

对于样例 1, $n = 5$:

- $a_1 = 1$;
- $a_1 - 2 = -1$, 不是正整数, 因此 $a_2 = a_1 + 2 = 3$;
- $a_2 - 3 = 0$, 不是正整数, 因此 $a_3 = a_2 + 3 = 6$;
- $a_3 - 4 = 2$, 是正整数, 且没有在数列中出现过, 因此 $a_4 = 2$;
- $a_4 - 5 = -3$, 不是正整数, 因此 $a_5 = a_4 + 5 = 7$;

a_1, a_2, a_3, a_4, a_5 从小到大排序后的结果为 1 2 3 6 7。

3.1.10 数据范围

对于所有数据点, 保证 $1 \leq n \leq 3000$ 。

3.1.11 参考程序

```
1 n = int(input())
2 a = [1]
3 for i in range(2, n + 1):
4     a.append(a[-1] - i if a[-1] - i > 0 and a[-1] - i not in a else a[-1] + i)
5
6 def bubble_sort(a):
7     n, flag = len(a), True
8     while flag:
9         flag = False
10        for i in range(n - 1):
11            if a[i] > a[i + 1]:
12                flag = True
13                a[i], a[i + 1] = a[i + 1], a[i]
14
15 bubble_sort(a)
16 for i in range(n):
17     a[i] = str(a[i])
18 print(' '.join(a))
```

3.2 编程题 2

- 试题名称: 字符排序
- 时间限制: 1.0 s
- 内存限制: 512.0 MB

3.2.1 题面描述

小杨有 n 个仅包含小写字母的字符串 s_1, s_2, \dots, s_n ，小杨想将这些字符串按一定顺序排列后拼接到一起构成字符串 t 。小杨希望最后构成的字符串 t 满足：

- 假设 t_i 为字符串 t 的第 i 个字符，对于所有的 $j < i$ 均有 $t_j \leq t_i$ 。两个字符的大小关系与其在字母表中的顺序一致，例如 $e < g < p < s$ 。

小杨想知道是否存在满足条件的字符串排列顺序。

3.2.2 输入格式

第一行包含一个正整数 T ，代表测试数据组数。

对于每组测试数据，第一行包含一个正整数 n ，含义如题面所示。

之后 n 行，每行包含一个字符串 s_i 。

3.2.3 输出格式

对于每组测试数据，如果存在满足条件的排列顺序，输出 1，否则输出 0。

3.2.4 样例

```
1 | 3
2 | 3
3 | aa
4 | ac
5 | de
6 | 2
7 | aac
8 | bc
9 | 1
10 | gesp
```

```
1 | 1
2 | 0
3 | 0
```

3.2.5 样例解释

对于第一组测试数据，一种可行的排列顺序为 $aa+ac+de$ ，构成的字符串 t 为 $aaacde$ ，满足条件。

对于全部数据，保证有 $1 \leq t, n \leq 100$ ，每个字符串的长度不超过 10。

3.2.6 参考程序

```
1 def bubble_sort(a):
2     n, flag = len(a), True
3     while flag:
4         flag = False
5         for i in range(n - 1):
6             if a[i] > a[i + 1]:
7                 flag = True
8                 a[i], a[i + 1] = a[i + 1], a[i]
9
10 t = int(input())
```

```
11 for _ in range(t):
12     n = int(input())
13     s = []
14     for _ in range(n):
15         s.append(input())
16     bubble_sort(s)
17     t = ''.join(s)
18     f1 = 1
19     for i in range(1, len(t)):
20         if t[i] < t[i - 1]:
21             f1 = 0
22             break
23     print(f1)
```