# Übungsklausur

Python 3 intoCODE

SoSe 2023

| **Surname** | **Vorname** | **Unterschrift** |
|---|---|---|
| | | |

| **Aufgabe** | **max. Punkte** | **erreicht** |
|---|---|---|
| 1 | 2.0 | |
| 2 | 3.0 | |
| 3 | 4.5 | |
| 4 | 3.0 | |
| 5 | 3.0 | |
| 6 | 3.0 | |
| 7 | 3.0 | |
| 8 | 3.0 | |
| 9 | 2.0 | |
| 10 | 4.0 | |
| 11 | 3.0 | |
| 12 | 3.0 | |
| 13 | 4.0 | |
| 14 | 2.0 | |
| 15 | 3.0 | |
| 16 | 2.5 | |
| 17 | 5.0 | |
| $\sum$ | **53** | |

1. (2.0 points) Welche Ausgaben liefert das folgende Programm?
   What will be the output of the following code?

```python
def f(x):
    x[2][0] = 15
    x[2] = x[0]
    return x

c = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
d = f(c)

print(f"{d[1] } {d[2]} {d[0] is d[1]}")
```

   A. [4, 5, 6] [1, 2, 3] False
   B. [1, 2, 3] [10, 8, 9] False
   C. [4, 5, 6] [10, 8, 9] True
   D. [7, 8, 9] [1, 2, 3] False

2. (3.0 points) Welche Ausgaben liefert das folgende Programm?
   What will be the output of the following code?

```python
def f(x):
 if x >= 1:
    return x*f(x - 1)
 else:
    return x

print(f(5//2))
```

   A. 1
   B. 0.50
   C. 0
   D. 1.875

3. (4.5 points) Was ist die Zeitkomplexität der folgenden Funktion?
   What is the complexity of the following function?

```python
def f(x):
    if x <= 1:
        return x
    else:
        return f(x-1) + f(x-2)
```

   A. $\mathcal{O}(2^n)$
   B. $\mathcal{O}(\log n)$
   C. $\mathcal{O}(n)$
   D. $\mathcal{O}(n^2)$

4. (3.0 points) Welche Ausgaben liefert das folgende Programm?
What will be the output of the following code?

```python
def f(x):
    if x <= 1:
        return x
    else:
        return f(x-1) + f(x-2)

print(f(8))
```

Fibonacci

A. 21

B. 13

C. 8

D. Error

5. (3.0 points) Was ist die Zeitkomplexität der folgenden Funktion?
What is the complexity of the following function?

```python
def f(x):
    n = len(x)
    for i in range(n):
        for j in range(n-i-1):
            if x[j] < x[j+1]:
                x[j], x[j+1] = x[j+1], x[j]
    return x
```

A. $\mathcal{O}(n \log n)$

B. $\mathcal{O}(2^n)$

C. $\mathcal{O}(n)$

D. $\mathcal{O}(n^2)$

E. $\mathcal{O}(\log n)$

6. (3.0 points) Welche der folgenden Möglichkeiten ist die richtige Art, eine Binärdatei in Python zu öffnen?
Which of the following is the correct way to open a binary file in Python?

A. `open(file, "r")`

B. `open(file, "w")`

C. `open(file, "a")`

D. `open(file, "b")`

7. (3.0 points) Welche Ausgaben liefert das folgende Programm?
What will be the output of the following code?

```
def f(x):
    n = len(x)
    for i in range(n):
        for j in range(n-i-1):
            if x[j] > x[j+1]:
                x[j], x[j+1] = x[j+1], x[j]
    return x

sample_list = [38,27,43,3,9,82,10]
print(f(sample_list))
```

A. [3, 9, 10, 27, 38, 43, 82]

B. [82, 43, 38, 27, 10, 9, 3]

C. [3, 9, 10, 82, 38, 43, 27]

D. [38, 27, 43, 3, 9, 82, 10]

8. (3.0 points) Was ist die Zeitkomplexität der folgenden Funktion?
What is the complexity of the following function?

```
def f(x):
    if len(x) < 2:
        return x

    p = x[0]

    l = [i for i in x[1:] if i>= p]
    r = [i for i in x[1:] if i<p]

    return  f(l) + [p] + f(r)
```

Bubble sort - O(n^2)

A. $\mathcal{O}(n \log n)$

B. $\mathcal{O}(n)$

C. $\mathcal{O}(n^2)$

D. $\mathcal{O}(\log n)$

9. (2.0 points) Welche Ausgaben liefert das folgende Programm?
What is the output of the following Python code?

```
x = 4
y = 1
print(y % x)
```

A. 0

B. 1

C. 2

D. 3

E. 5

10. (4.0 points) Welche Ausgaben liefert das folgende Programm?
What will be the output of the following code?

```python
def f(x):
    if len(x) < 2:
        return x
                                BUBBLE SORT - O(N^2)
    p = x[0]

    l = [i for i in x[1:] if i<= p]
    r = [i for i in x[1:] if i>p]

    return  f(l) + [p] + f(r)

x = [9, -3, 5, 2, 6, 8, -6, 1, 3]
print(f(x))
```

A. [9]

B. [9, -3, 5, 2, 6, 8, -6, 1, 3]

C. [9, 8, 6, 5, 3, 2, 1, -3, -6]

D. [-6, -3, 1, 2, 3, 5, 6, 8, 9]

11. (3.0 points) Welche Sortieralgorithmen verwenden den Ansatz "divide and conquer"?
Which sorting algorithms use the divide and conquer approach?

A. Merge Sort

B. Quick Sort

C. Merge Sort, Quick Sort

D. Bubble Sort, Merge Sort

E. HeapSort

12. (3.0 points) Was ist die Zeitkomplexität der folgenden Funktion?
What is the complexity of the following function?

```python
def f(n):
    if n <= 0:
        print("Countdown complete!")
    else:
        print(n)
        f(n - 1)
```

Linear - O (n)

```
10
9
8
7
6
5
4
3
2
1
Countdown complete!
```

A. $\mathcal{O}(n \log n)$

B. $\mathcal{O}(\log n)$

C. $\mathcal{O}(n)$

D. $\mathcal{O}(n^2)$

13. (4.0 points) Welche Ausgaben liefert das folgende Programm?
What will be the output of the following code?

```
def f(x):
    if len(x) == 1:
        return x[0]
    else:
        value = f(x[1:])
        return x[0] if x[0] > value else value

my_array = [9, -3, 5, 2, 6, 8, -6, 1, 3]
print(f(my_array))
```

A. 0

B. -6

C. 9

D. [-6, -3, 1, 2, 3, 5, 6, 8, 9]

14. (2.0 points) Welche Ausgaben liefert das folgende Programm?
What will be the output of the following code?

```
a = [x for x in range(10) if x%2==0]
print(a[-3])
```

A. 3

B. 2

C. 4

D. 5

15. (3.0 points) Geben Sie an, was von folgenden Python-Code-Zeilen ausgegeben wird:
What will be the output of the following code?

```
a = 1
b = "refugeeks"
c = [a, b]
a = 5%2
print(f"{c} {b} {a}")
```

A. [1, 'refugeeks'] refugeeks 2

B. [2.5, 'refugeeks'] refugeeks 2.5

C. [1, 'refugeeks'] refugeeks 1

D. [1, 'refugeeks'] refugeeks 2.5

16. (2.5 points) Welche Ausgaben liefert das folgende Programm?
What will be the output of the following code?

```
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = A[::-1]
print(B)
```

A[::-1]: Isso significa que você está criando uma nova lista ou matriz a partir dos elementos de A, mas com um passo de -1, que efetivamente percorre a lista de trás para frente

A. [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

B. [[7, 8, 9], [4, 5, 6], [1, 2, 3]]

C. [[7, 8, 9], [1, 2, 3]]

D. Error

17. (5.0 points) Welche Ausgaben liefert das folgende Programm?
What will be the output of the following code?

```
def f(x):
    def g(l, r):
        result = []
        i, j = 0, 0
        while i < len(l) and j < len(r):
            if l[i] < r[j]:
                result.append(l[i])
                i += 1
            else:
                result.append(r[j])
                j += 1
        result += l[i:]
        result += r[j:]
        return result

    if len(x) <= 1:
        return x
    m = len(x) // 2
    l = f(x[:m])
    r = f(x[m:])
    return g(l, r)

sample_list = [38,27,43,3,9,82,10]
results = f(sample_list)
print(results)
```

Merge Sort O (n log n)

A. [3, 9, 10, 27, 38, 43, 82]

B. [38, 27, 43, 3, 9, 82, 10]

C. [82, 43, 38, 27, 10, 9, 3]

D. [3, 9, 10, 82, 38, 43, 27]