Python Programming Examination

Guide: Shuvam Banerji Seal
Duration: 2 Hours
Maximum Marks: 50

Section A - Algorithm Development (25 Marks)

Write detailed algorithms for the following problems. Include input/output specifications, necessary data structures, and step-by-step solution approach.

1. (5 marks) **String Pattern Matcher**
   Design an algorithm that finds all occurrences of a pattern in a given string, where the pattern can contain special characters:

   - '?' matches any single character
   - '*' matches zero or more characters

   Example:

   ```
   Input:
   Text: "python programming"
   Pattern: "p*m*g"
   Output: True (matches "python programming")

   Input:
   Text: "hello world"
   Pattern: "he?lo"
   Output: True (matches "hello")
   ```

2. (5 marks) **Dictionary Merger**
   Design an algorithm to merge multiple dictionaries while handling conflicting keys according to these rules:

   - For numeric values, take the average
   - For strings, concatenate with a comma separator
   - For lists, combine unique elements maintaining order

   Example:

   ```
   Input:
   dict1 = {'a': 1, 'b': 'hello', 'c': [1, 2]}
   dict2 = {'a': 3, 'b': 'world', 'c': [2, 3]}
   Output:
   {'a': 2, 'b': 'hello,world', 'c': [1, 2, 3]}
   ```

3. (5 marks) **Tuple Matrix Operations**
   Design an algorithm to perform matrix operations using tuples as rows. The algorithm should support:

   - Matrix addition

- Matrix transposition
- Finding the diagonal elements

Example:

```
Input:
matrix1 = ((1, 2), (3, 4))
matrix2 = ((5, 6), (7, 8))
Operations: Add matrices, find diagonal
Output:
Sum: ((6, 8), (10, 12))
Diagonal: (1, 4)
```

4. (5 marks) **List Rotation with Conditions**
Design an algorithm to rotate a list based on these conditions:

- If element is even, rotate right by its value
- If element is odd, rotate left by its value
- Skip rotation if it would result in the same list

Example:

```
Input: [1, 2, 3, 4]
Process:
1 (odd) -> rotate left by 1: [2, 3, 4, 1]
2 (even) -> rotate right by 2: [3, 4, 1, 2]
Output: [3, 4, 1, 2]
```

5. (5 marks) **Mathematical Sequence Generator**
Design an algorithm to generate a sequence where each number is:

- Sum of previous three numbers if all are even
- Product of previous two numbers if both are odd
- Square of the previous number otherwise

Example:

```
Input: First three numbers: [2, 4, 6]
Output: [2, 4, 6, 12, 144, 20736, ...]
```

Section B - Python Concepts (20 Marks)

Answer the following questions about Python internals and implementation details. (2 marks each)

6. Explain the memory footprint difference between an empty list and an empty tuple in Python. Why does this difference exist?
7. How does Python's string interning mechanism work, and when does it automatically intern strings?
8. Describe the memory overhead of dictionary objects in Python. Why do dictionaries consume more memory than equivalent lists?
9. What is the time complexity of list slice operations in Python? How does memory allocation work during slicing?

10. Explain how Python’s garbage collector handles circular references between objects.
11. What are Python’s method resolution order (MRO) rules in multiple inheritance? Provide an example where C3 linearization matters.
12. How does Python implement its iterator protocol internally? Explain the relationship between `__iter__` and `__next__`.
13. Describe the difference between shallow and deep copying of nested data structures in Python. When would you use each?
14. What is the purpose of the `__slots__` attribute in Python classes? How does it affect memory usage?
15. Explain how Python’s context managers work internally. What methods must be implemented and when are they called?

Section C - String Formatting MCQs (5 Marks)

Choose the correct output for each string formatting operation. (1 mark each)

16. What is the output of:

```
name = "Python"
version = 3.9
print(f"{name:>10}␣{version:.1f}")
```

(a) `Python 3.9`
(b) ` Python 3.9`
(c) ` Python 4.0`
(d) `Python 3.9`

17. What is the output of:

```
num = 42
print("{:b}".format(num))
```

(a) `42`
(b) `101010`
(c) `0b101010`
(d) `0x2A`

18. What is the output of:

```
data = {’x’: 100, ’y’: 200}
print("%(x)d␣-␣%(y)d" % data)
```

(a) `x - y`
(b) `100 - 200`
(c) `{x} - {y}`
(d) `data[’x’] - data[’y’]`

19. What is the output of:

```
text = "Python"
print("{:.2}".format(text))
```

(a) `Py`
(b) `on`

(c) Python

(d) P

20. What is the output of:

```python
price = 42.1234
print(f"${price:>10,.2f}")
```

(a) `$42.12`

(b) `$ 42.12`

(c) ` $42.12`

(d) `$ 42.12`