Quiz 06–20: Lists and Sets

Name: _____ Date: _____

Instructions

- Answer in the blanks.
- For “write code” questions, write **valid Python code** (**no functions needed**).
- For “what does it print” questions, write the **exact** output (line by line).
- You may assume all inputs are valid (unless the question says otherwise).

Part A — Lists + For-loops (focus)

1) Build a list (write code)

Create a list named `items` that contains (in this order):

- the integer `7`
- the string `"hi"`
- the integer `0`
- the string `"bye"`

```
```

2) Index and update (what does it print?)

```
nums = [10, 3, 8, 3]
nums[1] = nums[1] * 2
nums[3] = nums[0] - nums[2]
print(nums)
```

Output:

```

```

3) `append` + `pop` (what does it print?)

```
letters = ["A", "B", "C"]
letters.append("D")
x = letters.pop()
letters.append("E")
print(x)
print(letters)
```

Output:

```

```

4) Double each number (fill in the blank)

Fill the blank so the code builds `doubles` as `[4, 10, 2, 8]`.

```
numbers = [2, 5, 1, 4]

doubles = []
for x in numbers:
    doubles.append(_____________________)

print(doubles)
```

5) Count items in a list (answer)

What is the output?

```
numbers = [6, 1, 8, 5, 9, 2, 7]

count = 0
for x in numbers:
    if x > 5:
        count = count + 1

print(count)
```

Output:

```

```

6) Find the largest value (write code)

Write code to print the **largest** number in the list.

```
numbers = [12, 5, 20, 8, 20, 3]

# your code here
```

Output:

```

```

7) Max value + first index (write code)

Write code to print:

- the max value, and
- the **first** index where it appears.

```
numbers = [4, 9, 2, 9, 1]

max_value = numbers[0]
max_index = 0

for i in range(len(numbers)):
    if _____________________________:
        max_value = ____________________
        max_index = ____________________

print(max_value)
print(max_index)
```

Output:

```

```

8) Filter out zeros (write code)

Build a new list `nonzeros` that keeps only the nonzero numbers (in the same order).

```
numbers = [0, 3, 0, 2, 5, 0, 1]

nonzeros = []

for x in numbers:
    if _______________________________:
        _______________________________

print(nonzeros)
```

Output:

```

```

9) Sum only the even numbers (write code)

Compute the sum of **even** numbers in the list.

```
numbers = [3, 4, 7, 2, 9, 10]

total = 0

for x in numbers:
    if _____________________________:
        _____________________________

print(total)
```

Output:

```

```

10) Nested loops over a list

Python code:

```
nums = [1, 2, 3, 4]

for i in range(len(nums)):
    for j in range(i + 1, len(nums)):
        print(nums[i], nums[j])
```

How many lines are printed? _____

11) Swap inside a list

What is the output?

```
a = [5, 9, 1, 8]

temp = a[0]
a[0] = a[3]
a[3] = temp

print(a)
```

Output:

```

```

12) Membership check (fill in the blanks)

Fill the blanks so the code prints **True** if `target` is in `numbers`, otherwise prints **False**.

```
numbers = [4, 7, 2, 9, 5]
target = 8

found = False
for x in numbers:
    if x == target:
        found = _____________________
        break

print(found)
```

Output:

```

```

Part B — Sets + List/Set synergy (focus)

13) List → set (what does it print?)

```
nums = [2, 3, 2, 5, 3, 7]
s = set(nums)
print(s)
print(len(s))
```

1) One possible printed set (order may vary): _____

2) `len(s) =` _____

14) Membership on a set (what does it print?)

```
fruits = {"apple", "banana", "pear"}
print("orange" in fruits)
print("pear" in fruits)
```

Output:

```

```

15) Set operations (write the results)

Let:

```
A = {1, 2, 3, 5}
B = {3, 4, 5, 6}
```

What are the resulting sets?

- `A | B` (union) = _____
- `A & B` (intersection) = _____
- `A - B` (in A but not in B) = _____
- `B - A` (in B but not in A) = _____

16) Count unique numbers (write code)

Write code to print how many **unique** numbers are in the list.

```
numbers = [1, 1, 2, 3, 3, 3, 5]

count = 0  # count unique numbers
seen = _____________________  # create an empty set

for x in numbers:
    if x ________ seen:  # in or not in?
        seen._______________  # add x to seen
        count = ____________

print(count)
```

Output:

```
```

17) All unique? (write code)

Write code that prints `True` if **all** items in `letters` are unique, else prints `False`.

```
letters = ["A", "B", "C", "B"]

seen = _________________  # create an empty set
unique = _______________  # True or False?

for x in letters:
    if x in seen:
        unique = _______________  # True or False?
        __________________________  # stop the loop
    seen._____________________  # add x to seen

print(unique)
```

18) Remove duplicates but keep order (write code)

Build a new list `unique_words` that keeps the **first** time each word appears.

Example:

- input: `["to", "be", "or", "not", "to", "be"]`
- output: `["to", "be", "or", "not"]`

```
words = ["to", "be", "or", "not", "to", "be"]

unique_words = ___________________________  # create an empty list
seen = ____________________________________  # create an empty set

for w in words:
    if w __________ seen:  # in or not in?
        unique_words.____________________
        seen._____________________________

print(unique_words)
```

19) Common items from two lists (write code)

Print the set of items that appear in **both** lists. Order may vary.

Hint:

- Convert the two lists to sets.
- The intersection of two sets **contains** items that appear in both lists.

```
a = ["red", "blue", "blue", "green"]
b = ["green", "yellow", "blue"]

# your code here
```

---

20) Filter using a “banned” set (write code)

Build a list named `allowed` (in the same order as numbers). Keep numbers that are **not** banned.

```
numbers = [5, 2, 9, 2, 7, 1, 9]
banned = {2, 9}

allowed = _____________________  # create an empty list

for x in numbers:
    if ____________________________:
        allowed._________________

print(allowed)
```

Output:

```

```

Part C — Number loops + Two Sum + Complexity

21) Factors into a list (write code)

Build a list `factors` that contains all positive factors of `n` in increasing order.

```
n = 18

factors = _____________________  # create an empty list

for i in range(______, ________):
    if ____________________________:
        ____________________________

print(factors)
```

Output:

```

```

22) Sum of digits (while-loop) (write code)

Compute the sum of digits of `n`.

Example: `n = 2405` → sum is `2 + 4 + 0 + 5 = 11`.

```
n = 2405

total = 0

while n > 0:
    x = ____________________  # get the last digit
    total = ________________
    n = ____________________  # remove the last digit from n

print(total)
```

23) Count digits (while-loop) (write code)

Count how many digits are in `n`. Assume `n` is a positive integer.

Example: `n = 2026` → answer is `4`.

```
n = 90700

count = 0

while _________________:
    count = _________________
    n = ____________________

print(count)
```