Quiz 08–12: For-Loops and While-Loops

Part A — For-loop with lists (traversal + accumulators)

1) Double each number

Output:

```
4
10
2
8
```

2) Running total

Fill in:

```
total = total + x
```

(Complete code)

```
numbers = [3, 1, 4]

total = 0
for x in numbers:
    total = total + x
    print(total)
```

3) Indentation fix

Corrected code:

```
numbers = [9, 8, 7]
for x in numbers:
    print(x)
```

4) Build a new list: squares

One correct solution:

```
nums = [1, 3, 5, 2]
squares = []

for x in nums:
    squares.append(x * x)

print(squares)
```

Output:

```
[1, 9, 25, 4]
```

Part B — `break` and searching

5) First multiple of 3

Output:

```
not yet: 4
first multiple of 3: 6
```

---

6) Find an index

Fill in:

- `idx = _____` → `idx = i`
- `___________` → `break`

(Complete code)

```
nums = [10, 20, 30, 40]
target = 30

idx = -1
for i in range(len(nums)):
    if nums[i] == target:
        idx = i
        break
print(idx)
```

---

7) Stop when the total gets too big

One correct solution:

```
nums = [2, 4, 5, 3]

total = 0

for x in nums:
    total = total + x
    print(total)
    if total > 10:
        break

print(total)
```

Output:

```
2
6
11
11
```

8) First even number

Output:

```
first even: 8
```

Part C — Number problems (`%` and `//`)

9) Last digit and remove last digit

Output:

```
7
50
0
```

10) Sum of digits

Fill in:

```
while n > 0:
```

One correct solution:

```
n = 2305
total = 0

while n > 0:
    x = n % 10
    total = total + x
    n = n // 10

print(total)
```

Output:

```
10
```

11) Factors by hand

```
1, 2, 3, 4, 6, 8, 12, 24
```

12) Print factors, but stop early

Fill in:

- `range(_____, _____)` → `range(1, n + 1)`
- `if _____:` → `if n % i == 0:`
- `count = _____` → `count = count + 1`
- `if _____:` → `if count == 4:`

(Complete code)

```
n = 30
count = 0

for i in range(1, n + 1):
    if n % i == 0:  # i is a factor of n
        print(i)
        count = count + 1
    if count == 4:
        break
```

Output:

```
1
2
3
5
```

13) Prime or not?

```
True
```

Part D — Nested loops and 2D lists

14) Pairs

Output:

```
A 1
A 2
A 3
B 1
B 2
B 3
```

15) Counting prints

```
6
```

16) Sum all numbers in a 2D list

Fill in:

- Outer loop: `for row in grid:`
- Inner loop: `for x in row:`
- Update: `total = total + x`

(Complete code)

```
grid = [
    [1, 2, 3],
    [4, 5, 6]
]

total = 0

for row in grid:
    for x in row:
        total = total + x

print(total)
```

Output:

```
21
```

17) Build a small multiplication table

Fill in:

- `for i in range(_____, _____)` → `for i in range(1, 4)`
- `for j in range(_____, _____)` → `for j in range(1, 5)`
- `table.append(_____)` → `table.append(row)`

(Complete code)

```
table = []

for i in range(1, 4):  # row i
    row = []
    for j in range(1, 5):  # column j
        s = str(i) + " x " + str(j) + " = " + str(i * j)
        row.append(s)
    table.append(row)  # add the row to table

for row in table:
    print(row)
```

18) Print a rectangle of stars

One correct solution:

```
for _ in range(3):
    print("*****")
```

Output:

```
*****
*****
*****
```

Part E — Challenges (mix topics)

19) Digit sums for a list

Fill in:

- `x = _____` → `x = k % 10`
- `k = _____` → `k = k // 10`
- `total = _____` → `total = total + x`

(Complete code)

```
nums = [12, 305, 40, 7]
digit_sums = []

for n in nums:
    total = 0
    k = n
    while k > 0:
        x = k % 10      # get the last digit of k
        k = k // 10     # remove the last digit of k
        total = total + x

    digit_sums.append(total)

print(digit_sums)
```

20) Max digit sum

Fill in:

- `while k ___ 0:` → `while k > 0:`
- `x = _____` → `x = k % 10`
- `k = _____` → `k = k // 10`
- `total = _____` → `total = total + x`
- `if _____:` → `if s > m:`
- `_____` → `m = s`

(Complete code)

```
nums = [91, 28, 100, 47]

digit_sums = []
for n in nums:
    total = 0
    k = n
    while k > 0:
        x = k % 10
        k = k // 10
        total = total + x
    digit_sums.append(total)

m = digit_sums[0]
for s in digit_sums:
    if s > m:
        m = s

print(m)
```

Output:

```
11
```

21) Search a 2D list and stop early

Fill in:

- `if _____:` → `if n == target:`
- `found = _____` → `found = True`
- inner stop: `break`
- outer stop: `break`

(Complete code)

```
grid = [
    [5, 1, 9],
    [2, 7, 3],
    [4, 6, 8]
]
target = 7

found = False

for row in grid:
    for n in row:
        if n == target:
            found = True
            break  # stop the inner loop
    if found:
        break  # stop the outer loop

if found:
    print("Found")
else:
    print("Not found")
```

Output:

```
Found
```

---

22) Fibonacci list + count

Fill in:

- `while _____:`
  - `while fib[-1] + fib[-2] <= 100:`
- `fib.append(_____)`
  - `fib.append(fib[-1] + fib[-2])`

(Complete code)

```
fib = [1, 1]

while fib[-1] + fib[-2] <= 100:
    fib.append(fib[-1] + fib[-2])

print(fib)
print(len(fib))
```

Output:

```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
11
```

---

23) Two Sum (nested loops + `break`)

Fill in (one correct set):

- `found = _____` → `found = False`
- `for i in range(_____, _____):` → `for i in range(0, n):`
- `for j in range(_____, _____):` → `for j in range(i + 1, n):`
- `if _____:` → `if nums[i] + nums[j] == target:`
- `found = _____` → `found = True`
- stop inner: `break`
- `if _____:` → `if found:`
- stop outer: `break`

(Complete code)

```
nums = [4, 1, 9, 2, 7]
target = 11

n = len(nums)
found = False

for i in range(0, n):
    for j in range(i + 1, n):
        if nums[i] + nums[j] == target:
            found = True
            break

    if found:
        break

print(found)
```

Output:

```
True
```