Quiz 14–17: While-Loops

Part A — While-loop basics

1) While-loop facts (True/False)

1. **True**
2. **True**
3. **True**

2) Count down (what does it print?)

Output:

```
5
3
1
done
```

3) Fix the bug (infinite loop)

Corrected line:

```
x = x - 1
```

4) First number bigger than 10 (what does it print?)

Output:

```
not yet: 2
not yet: 5
not yet: 8
found: 11
```

5) Count jumps by 4 (write code)

One correct completion (matches the example that **counts the jump that passes** `limit`):

```
start = 3
limit = 20

count = 0

while True:
    if start > limit:
        break
    start = start + 4
    count = count + 1

print(count)
```

Part B — Digits (using `//` and `%`)

6) Count digits (what does it print?)

Output:

```
4
```

7) Last digit + remove last digit (fill in the blanks)

Blanks:

- `% 10`
- `// 10`

Completed code:

```
n = 735

while n > 0:
    last = n % 10
    print(last)
    n = n // 10
```

Output:

```
5
3
7
```

8) Sum of digits (fill in the blanks)

Blanks:

- `% 10`
- `digit`
- `// 10`

Completed code:

```
n = 409
total = 0

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

print(total)
```

Output:

```
13
```

9) Collatz game (what does it print?)

Output:

```
7
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1
```

10) Factors using `while` (write code)

One correct completion:

```
n = 12

i = 1

while i <= n:
    if n % i == 0:
        print(i)
    i = i + 1
```

Output:

```
1
2
3
4
6
12
```

Part C — Palindrome with two pointers (strings)

11) Palindrome facts (True/False)

1. **True**
2. **True**
3. **True**

---

12) Trace the pointers (fill in)

- First comparison: `s[left]` is **"r"** and `s[right]` is **"r"**
- After one successful match: `left` becomes **1** and `right` becomes **5**
- Second comparison: `s[left]` is **"a"** and `s[right]` is **"a"**

---

13) Read code and reason (what does it print?)

Output:

```
True
False
```

---

14) Palindrome checker (write code)

One correct completion:

```
s = "madam"

left = 0
right = len(s) - 1
palindrome = True

while left < right:
    if s[left] != s[right]:
        palindrome = False
        break

    left = left + 1
    right = right - 1

if palindrome:
    print("YES")
else:
    print("NO")
```

Part D — Swap and reverse

15) Swap two numbers (fill in the blanks)

Blanks:

- `a`
- `b`
- `temp`

Completed code:

```
a = 6
b = 7

temp = a
a = b
b = temp

print(a, b)
```

Output:

```
7 6
```

16) Reverse a list in-place (what does it print?)

Output:

```
[5, 4, 3, 2, 1]
```

17) Fix the reverse bug (one line)

Corrected line:

```
right = right - 1
```

Part E — Move zeros to the end

18) Fill in the blanks: move zeros in-place

Blanks:

- `write = 0`
- `read = 0`
- `numbers[write] = numbers[read]`
- `write = write + 1`
- `read = read + 1`
- `while k < len(numbers):`
- `numbers[k] = 0`
- `k = k + 1`

Completed code:

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

write = 0
read = 0

# Step 1: copy nonzeros forward
while read < len(numbers):
    if numbers[read] != 0:
        numbers[write] = numbers[read]
        write = write + 1
    read = read + 1

# Step 2: fill zeros from write to the end
k = write
while k < len(numbers):
    numbers[k] = 0
    k = k + 1

print(numbers)
```

Output:

```
[1, 3, 2, 0, 0, 0]
```