Worksheet 11: Python Nested For-Loop

Name: _____ Date: _____

Instructions

- Answer in the blanks.
- For “write code” questions, write valid Python code.
- For “what does it print” questions, write the exact output.

Part A — Nested loop basics

1) Review: list traversal

What does this print?

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

for x in numbers:
    print(x)
```

Output:

```

```

2) What does this print? (nested loop)

```
for outer in [1, 2]:
    for inner in ["A", "B"]:
        print(outer, inner)
```

Output:

```

```

3) How many times does `print` run?

```
count = 0

for i in range(3):
    for j in range(4):
        count = count + 1

print(count)
```

Output:

```

```

Part B — 2D lists (list of lists)

4) 2D list print

What does this print?

```
grid = [
    [10, 20],
    [30, 40],
]

for row in grid:
    for x in row:
        print(x)
```

Output:

```

```

5) Flatten a 2D list

Complete the code:

```
grid = [
    [10, 20],
    [30, 40],
]

flat = []

for row in grid:
    for x in row:
        ________________________

print(flat)
```

Expected output:

```
[10, 20, 30, 40]
```

6) Sum all numbers in a 2D list

Complete the code to print the sum of all numbers in `grid`.

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

total = 0

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

print(total)
```

Output:

```

```

7) Count items in a 2D list

Complete the code to count how many numbers are inside `grid`.

```
grid = [
    [9, 8, 7],
    [6, 5],
    [4],
]

count = 0

for row in grid:
    for x in row:
        _________________________

print(count)
```

Output:

```

```

Part C — Cartesian product (all pairs)

8) Build pairs like "X1", "X2", ...

The goal is to make:

```
['X1', 'X2', 'X3', 'Y1', 'Y2', 'Y3']
```

Complete the code:

```
letters = ["X", "Y"]
numbers = [1, 2, 3]  # integers

result = []

for l in _____________________:
    for n in _____________________:
        result.append(_____ + _____)

print(result)
```

9) Grid coordinates (strings)

We want a list like this:

```
['(0,0)', '(0,1)', '(0,2)', '(1,0)', '(1,1)', '(1,2)']
```

Complete the code:

```
coords = []

for r in range(2):        # r = 0, 1
    for c in range(3):    # c = 0, 1, 2
        coords.append("(" + str(r) + "," + _____________________ + ")")

print(coords)
```

10) Make a small multiplication table (as a 2D list)

What does this print?

```
table = []

for i in range(1, 4):      # 1, 2, 3
    row = []
    for j in range(1, 4):  # 1, 2, 3
        row.append(i * j)
    table.append(row)

print(table)
```

Output:

```

```

Part D — Try all pairs (Two Sum idea)

11) Two Sum: does a pair exist?

What does this print?

```
numbers = [2, 15, 7, 8]
target = 9

n = len(numbers)
found = False

for i in range(n - 1):
    for j in range(i + 1, n):
        if numbers[i] + numbers[j] == target:
            found = True

print(found)
```

Output:

```
```

12) Two Sum: stop early using `break`

Complete the code so it stops early when the pair is found.

```
numbers = [2, 15, 7, 8]
target = 9

n = len(numbers)
found = False

for i in range(n - 1):
    for j in range(i + 1, n):
        if numbers[i] + numbers[j] == target:
            found = True
            _____________________   # break inner loop
    if found:
        _____________________       # break outer loop

print(found)
```

13) List all pairs checked (order matters)

For the code below, list the pairs **in order** as `(numbers[i], numbers[j])`.

```
numbers = [4, 1, 9, 3]

n = len(numbers)

for i in range(n - 1):
    for j in range(i + 1, n):
        print(numbers[i], numbers[j])
```

Write the output pairs in order:

1. _____
2. _____
3. _____

4. _____
5. _____
6. _____

Part E — Nested loops for patterns

14) Print a rectangle of stars

Complete the code (hint: build a `line` string, then print it):

```
for r in range(3):
    line = ""
    for c in range(5):
        line = line + _____________________
    print(line)
```

Expected output: 3 rows, and each row has 5 stars.

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