# Worksheet 15: Palindrome Check with Two Pointers

Name: _____ Date: _____

## Instructions

- Answer in the blanks.
- For “write code” questions, write valid Python code.
- For “what is the output” questions, write the exact output (including spaces/newlines).

## Part A — Palindrome basics

### 1) Which strings are palindromes?

Circle **True** or **False** for each string.

- `"level"` True / False
- `"robot"` True / False
- `"ABBA"` True / False
- `""` True / False
- `"A"` True / False

### 2) Fill in the blanks: the two pointers idea

First understand the code. Then fill in the blanks.

```
s = "ABCBA"

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

while left < right:
    if s[left] != s[right]: # mismatch
        palindrome = False
        break
    left = left + 1
    right = right - 1

print(palindrome)
```

Fill in the blanks:

- `left` starts at _____
- `right` starts at _____
- We keep looping while _____.

Inside the loop:

- If `s[left] != s[right]`, then `palindrome = _____`.
- If they match, we do:
  - `left = left ____ 1`
  - `right = right ____ 1`

---

# Part B — Trace the loop

## 3) Trace the output: `"ABCBA"`

What does this code print?

```
s = "ABCBA"

left = 0
right = len(s) - 1

while left < right:
    print(left, right, s[left], s[right])
    if s[left] != s[right]:
        break
    left = left + 1
    right = right - 1
```

Output:

```

```

## 4) Trace the output: `"ABCDBA"`

What does this code print?

```
s = "ABCDBA"

left = 0
right = len(s) - 1

while left < right:
    print(left, right, s[left], s[right])
    if s[left] != s[right]:
        break
    left = left + 1
    right = right - 1
```

Output:

# Part C — Read code and reason

## 5) Empty string

What does this code print?

```
s = ""

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

while left < right:
    if s[left] != s[right]:
        palindrome = False
        break
    left = left + 1
    right = right - 1

print(palindrome)
```

Output:

## 6) Debugging (2 bugs)

This code is **trying** to check whether `s` is a palindrome.

Line A and B are **wrong**. Write the **correct version** of those lines below.

```
s = "ABBA"

palindrome = True
left = 0
right = len(s)      # Line A

while left < right:
    if s[left] != s[right]:
        palindrome = False
        break
    left = left + 1
    right = right + 1  # Line B

print(palindrome)
```

Corrected lines:

- Line A: `right = _____`
- Line B: `right = _____`

---

# Part D — Write code

## 7) Fill in the missing code

Fill in the blanks to complete a palindrome checker.

```
s = "racecar"

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

while left < right:
    if s[left] != s[right]:
        palindrome = _____________________
        _____________________  # stop the loop
    left = _____________________
    right = _____________________

print(palindrome)
```

---

## 8) Print the first mismatch (if any)

Write code that prints the **first mismatching pair** of characters and then stops.

Use this input:

```
s = "ABCDXDCBA"
```

Rules:

- If you find a mismatch, print the two characters (example output: `C D`) and stop.
- If there is **no mismatch**, print `No mismatch`.

Complete the code:

```
s = "ABCDXDCBA"

left = 0
right = len(s) - 1
found = False

while _____________________________:
    if _____________________________:
        print(_____________________)
        found = ___________
        _____________________  # stop the loop
    left = ____________________
    right = ____________________

if ____________________:
    print("No mismatch")
```

Expected output:

```
C D
```