"""

Python Worksheet with Solutions

==========

This worksheet covers the following topics:

1. Conditional blocks using if, else, and elif

2. Simple for loops in Python, for loops using ranges, and the use of while and do-while loops in Python

3. Loop manipulation using pass, continue, break, and else

Each topic includes problems of varying complexity: Simple, Intermediate, and Complex.

==========

Topic 1: Conditional Blocks using if, else, and elif

==========

Simple Problems

Problem 1: Check if a number is positive, negative, or zero.

```
print("Problem 1: Check if a number is positive, negative, or zero.")
num = 10
if num > 0:
    print("Positive")
elif num < 0:
    print("Negative")
else:
    print("Zero")
print("-" * 50)
```

Problem 2: Determine if a year is a leap year.

```
print("Problem 2: Determine if a year is a leap year.")
year = 2000
if (year % 4) != 0:
   print(False)
elif (year % 100) != 0:
   print(True)
elif (year % 400) != 0:
   print(False)
else:
   print(True)
print("-" * 50)

# Problem 3: Check if a character is a vowel or consonant.
print("Problem 3: Check if a character is a vowel or consonant.")
char = 'b'
vowels = 'aeiouAEIOU'
if char in vowels:
   print("Vowel")
elif char.isalpha():
   print("Consonant")
else:
   print("Not an alphabet")
print("-" * 50)

# Problem 4: Determine the largest of three numbers.
print("Problem 4: Determine the largest of three numbers.")
a, b, c = 10, 20, 15
if a >= b and a >= c:
   print(a)
elif b >= a and b >= c:
   print(b)
```

```
else:
    print(c)
print("-" * 50)

# Problem 5: Check if a number is even or odd.
print("Problem 5: Check if a number is even or odd.")
num = 7
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
print("-" * 50)

# Problem 6: Determine if a person is eligible to vote based on age.
print("Problem 6: Determine if a person is eligible to vote based on age.")
age = 16
if age >= 18:
    print("Eligible to vote")
else:
    print("Not eligible to vote")
print("-" * 50)

# Intermediate Problems

# Problem 7: Calculate grade based on marks.
print("Problem 7: Calculate grade based on marks.")
marks = 85
if marks >= 90:
    grade = "A"
elif marks >= 80:
    grade = "B"
```

```
elif marks >= 70:
    grade = "C"
elif marks >= 60:
    grade = "D"
else:
    grade = "F"
print(grade)
print("-" * 50)

# Problem 8: Determine the type of triangle based on side lengths.
print("Problem 8: Determine the type of triangle based on side lengths.")
a, b, c = 3, 4, 4
if a == b == c:
    triangle = "Equilateral"
elif a == b or b == c or a == c:
    triangle = "Isosceles"
else:
    triangle = "Scalene"
print(triangle)
print("-" * 50)

# Problem 9: Check if a year is a century year.
print("Problem 9: Check if a year is a century year.")
year = 1900
if year % 100 == 0:
    print("Century Year")
else:
    print("Not a Century Year")
print("-" * 50)

# Problem 10: Determine the category of a person based on age.
```

```
print("Problem 10: Determine the category of a person based on age.")
age = 45
if age < 13:
    category = "Child"
elif age < 20:
    category = "Teenager"
elif age < 65:
    category = "Adult"
else:
    category = "Senior"
print(category)
print("-" * 50)

# Problem 11: Calculate the absolute value of a number.
print("Problem 11: Calculate the absolute value of a number.")
num = -5
if num >= 0:
    abs_val = num
else:
    abs_val = -num
print(abs_val)
print("-" * 50)

# Problem 12: Check if a string starts with a vowel.
print("Problem 12: Check if a string starts with a vowel.")
s = "Apple"
vowels = 'aeiouAEIOU'
if s and s[0] in vowels:
    print("Starts with a vowel")
else:
    print("Does not start with a vowel")
```

```
print("-" * 50)

# Complex Problems

# Problem 13: Determine if a number is prime.
print("Problem 13: Determine if a number is prime.")
num = 29
if num <= 1:
    is_prime = False
elif num <= 3:
    is_prime = True
elif num % 2 == 0 or num % 3 == 0:
    is_prime = False
else:
    i = 5
    is_prime = True
    while i * i <= num:
        if num % i == 0 or num % (i + 2) == 0:
            is_prime = False
            break
        i += 6
print(is_prime)
print("-" * 50)

# Problem 14: Find the greatest common divisor (GCD) of two numbers.
print("Problem 14: Find the greatest common divisor (GCD) of two numbers.")
a, b = 48, 18
while b:
    a, b = b, a % b
gcd = a
print(gcd)
```

```
print("-" * 50)

# Problem 15: Check if a string is a palindrome.
print("Problem 15: Check if a string is a palindrome.")
s = "A man a plan a canal Panama"
filtered_s = ''.join(filter(str.isalnum, s)).lower()
if filtered_s == filtered_s[::-1]:
    print("Palindrome")
else:
    print("Not a palindrome")
print("-" * 50)

# Problem 16: Determine if three angles can form a valid triangle.
print("Problem 16: Determine if three angles can form a valid triangle.")
angle1, angle2, angle3 = 60, 60, 60
if angle1 + angle2 + angle3 == 180 and angle1 > 0 and angle2 > 0 and angle3 > 0:
    print("Valid Triangle")
else:
    print("Invalid Triangle")
print("-" * 50)

# Problem 17: Calculate the factorial of a number using conditional statements.
print("Problem 17: Calculate the factorial of a number using conditional statements.")
n = 5
if n < 0:
    print("Undefined for negative numbers")
elif n == 0 or n ==1:
    print(1)
else:
    result = 1
    for i in range(2, n+1):
```

```
    result *= i
  print(result)
print("-" * 50)

# Problem 18: Determine the type of a quadrilateral based on side lengths.
print("Problem 18: Determine the type of a quadrilateral based on side lengths.")
a, b, c, d = 2, 2, 2, 2
if a == b == c == d:
  quad = "Square"
elif a == c and b == d:
  quad = "Rectangle"
elif a == b or b == c or c == d or d == a:
  quad = "Rhombus or Other"
else:
  quad = "General Quadrilateral"
print(quad)
print("-" * 50)

================================================================================
Topic 2: Loops in Python
================================================================================

# Simple Problems

# Problem 19: Print numbers from 1 to 5 using a for loop.
print("Problem 19: Print numbers from 1 to 5 using a for loop.")
for i in range(1, 6):
  print(i)
print("-" * 50)
```

```
# Problem 20: Calculate the sum of first 10 natural numbers using a for loop.
print("Problem 20: Calculate the sum of first 10 natural numbers using a for loop.")
total = 0
for i in range(1, 11):
    total += i
print(total)
print("-" * 50)

# Problem 21: Print each character in a string using a for loop.
print("Problem 21: Print each character in a string using a for loop.")
s = "Hello"
for char in s:
    print(char)
print("-" * 50)

# Problem 22: Use a while loop to print numbers from 1 to 5.
print("Problem 22: Use a while loop to print numbers from 1 to 5.")
i = 1
while i <= 5:
    print(i)
    i += 1
print("-" * 50)

# Problem 23: Find the product of numbers from 1 to 4 using a for loop.
print("Problem 23: Find the product of numbers from 1 to 4 using a for loop.")
product = 1
for i in range(1, 5):
    product *= i
print(product)
print("-" * 50)
```

```
# Problem 24: Print the first 5 even numbers using a for loop.
print("Problem 24: Print the first 5 even numbers using a for loop.")
for i in range(2, 11, 2):
    print(i)
print("-" * 50)

# Intermediate Problems

# Problem 25: Calculate the factorial of a number using a for loop.
print("Problem 25: Calculate the factorial of a number using a for loop.")
n = 6
factorial = 1
for i in range(2, n+1):
    factorial *= i
print(factorial)
print("-" * 50)

# Problem 26: Print the Fibonacci sequence up to n terms using a for loop.
print("Problem 26: Print the Fibonacci sequence up to n terms using a for loop.")
n = 7
a, b = 0, 1
sequence = []
for _ in range(n):
    sequence.append(a)
    a, b = b, a + b
print(sequence)
print("-" * 50)

# Problem 27: Find the sum of all even numbers in a list using a for loop.
print("Problem 27: Find the sum of all even numbers in a list using a for loop.")
lst = [1, 2, 3, 4, 5, 6]
```

```
total = 0
for num in lst:
    if num % 2 == 0:
        total += num
print(total)
print("-" * 50)

# Problem 28: Use a for loop with range to print multiples of 3 up to 30.
print("Problem 28: Use a for loop with range to print multiples of 3 up to 30.")
for i in range(3, 31, 3):
    print(i)
print("-" * 50)

# Problem 29: Reverse a string using a for loop.
print("Problem 29: Reverse a string using a for loop.")
s = "Python"
reversed_s = ""
for char in s:
    reversed_s = char + reversed_s
print(reversed_s)
print("-" * 50)

# Problem 30: Use a while loop to find the first number greater than 100 divisible by 7.
print("Problem 30: Use a while loop to find the first number greater than 100 divisible by 7.")
num = 101
while True:
    if num % 7 == 0:
        print(num)
        break
    num += 1
print("-" * 50)
```

```
# Complex Problems


# Problem 31: Implement a simple do-while loop using a while loop to prompt user input until 'exit' is entered.
print("Problem 31: Implement a simple do-while loop simulation.")
# Uncomment the lines below to run the interactive function
# while True:
#     user_input = input("Enter something (type 'exit' to quit): ")
#     if user_input.lower() == 'exit':
#         break
#     print("You entered:", user_input)
print("Interactive function skipped.")
print("-" * 50)

# Problem 32: Print a diamond pattern using nested for loops.
print("Problem 32: Print a diamond pattern using nested for loops.")
n = 3
# Upper part
for i in range(n):
    print(' ' * (n - i - 1) + '*' * (2 * i + 1))
# Lower part
for i in range(n-2, -1, -1):
    print(' ' * (n - i - 1) + '*' * (2 * i + 1))
print("-" * 50)

# Problem 33: Generate a multiplication table for a given number using nested loops.
print("Problem 33: Generate a multiplication table for a given number using nested loops.")
num = 5
upto = 10
for i in range(1, upto + 1):
```

```
    print(f"{num} x {i} = {num * i}")
print("-" * 50)

# Problem 34: Find all prime numbers up to a given number using loops.
print("Problem 34: Find all prime numbers up to a given number using loops.")
n = 20
primes = []
for num in range(2, n + 1):
    is_prime = True
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            is_prime = False
            break
    if is_prime:
        primes.append(num)
print(primes)
print("-" * 50)

# Problem 35: Implement a nested loop to print a matrix.
print("Problem 35: Implement a nested loop to print a matrix.")
rows, cols = 2, 3
for i in range(rows):
    for j in range(cols):
        print(f"{i},{j}", end=' ')
    print()
print("-" * 50)

# Problem 36: Calculate the sum of digits of a number using loops.
print("Problem 36: Calculate the sum of digits of a number using loops.")
num = 12345
total = 0
```

```
while num > 0:
    total += num % 10
    num = num // 10
print(total)
print("-" * 50)

================================================================================
Topic 3: Loop Manipulation using pass, continue, break, and else
================================================================================

# Simple Problems

# Problem 37: Use 'continue' to skip even numbers in a loop.
print("Problem 37: Use 'continue' to skip even numbers in a loop.")
for i in range(1, 6):
    if i % 2 == 0:
        continue
    print(i)
print("-" * 50)

# Problem 38: Use 'break' to exit a loop when a condition is met.
print("Problem 38: Use 'break' to exit a loop when a condition is met.")
for i in range(1, 11):
    if i == 5:
        break
    print(i)
print("-" * 50)

# Problem 39: Use 'pass' in a loop as a placeholder.
print("Problem 39: Use 'pass' in a loop as a placeholder.")
```

```
for i in range(1, 4):
    if i == 2:
        pass  # Do nothing
    print(i)
print("-" * 50)

# Problem 40: Use 'else' with a for loop when no break occurs.
print("Problem 40: Use 'else' with a for loop when no break occurs.")
for i in range(3):
    print(i)
else:
    print("Completed without break")
print("-" * 50)

# Problem 41: Use 'else' with a for loop when a break occurs.
print("Problem 41: Use 'else' with a for loop when a break occurs.")
for i in range(5):
    if i == 3:
        break
    print(i)
else:
    print("Completed without break")
print("-" * 50)

# Problem 42: Use 'continue' to skip certain iterations.
print("Problem 42: Use 'continue' to skip certain iterations.")
for i in range(1, 7):
    if i % 3 == 0:
        continue
    print(i)
print("-" * 50)
```

```
# Intermediate Problems

# Problem 43: Find the first occurrence of a number in a list and break the loop.
print("Problem 43: Find the first occurrence of a number in a list and break the loop.")
lst = [1, 2, 3, 4, 5]
target = 3
for num in lst:
    if num == target:
        print(f"Found {target}")
        break
else:
    print(f"{target} not found")

target = 6
for num in lst:
    if num == target:
        print(f"Found {target}")
        break
else:
    print(f"{target} not found")
print("-" * 50)

# Problem 44: Use 'pass' in exception handling within a loop.
print("Problem 44: Use 'pass' in exception handling within a loop.")
lst = ['a', 'skip', 'b', 'skip', 'c']
for item in lst:
    try:
        print(f"Processing {item}")
        if item == 'skip':
            raise ValueError
```

```
    except ValueError:
        pass  # Ignore the error and continue
print("-" * 50)

# Problem 45: Use 'continue' and 'break' in nested loops.
print("Problem 45: Use 'continue' and 'break' in nested loops.")
for i in range(1, 4):
    for j in range(1, 4):
        if j == 2:
            continue
        if i == 3 and j == 3:
            break
        print(f"i={i}, j={j}")
else:
    print("Completed all loops")
print("-" * 50)

# Problem 46: Use 'else' with a while loop.
print("Problem 46: Use 'else' with a while loop.")
count = 0
while count < 3:
    print(count)
    count += 1
else:
    print("While loop completed without break")
print("-" * 50)

# Problem 47: Use 'break' in a while loop when a condition is met.
print("Problem 47: Use 'break' in a while loop when a condition is met.")
count = 0
while True:
```

```
    if count == 4:
        break
    print(count)
    count += 1
print("-" * 50)

# Problem 48: Use 'continue' in a while loop to skip certain iterations.
print("Problem 48: Use 'continue' in a while loop to skip certain iterations.")
count = 0
while count < 5:
    count += 1
    if count == 3:
        continue
    print(count)
print("-" * 50)

# Complex Problems

# Problem 49: Implement a menu-driven program using loops and conditional statements with loop controls.
print("Problem 49: Implement a menu-driven program simulation.")
# Uncomment the lines below to run the interactive function
# while True:
#     print("\nMenu:")
#     print("1. Say Hello")
#     print("2. Add two numbers")
#     print("3. Exit")
#     choice = input("Enter your choice: ")
#     if choice == '1':
#         print("Hello!")
#     elif choice == '2':
```

```
#       try:
#           num1 = float(input("Enter first number: "))
#           num2 = float(input("Enter second number: "))
#           print(f"Sum: {num1 + num2}")
#       except ValueError:
#           print("Invalid input. Please enter numbers.")
#    elif choice == '3':
#       print("Exiting...")
#       break
#    else:
#       print("Invalid choice. Please try again.")
print("Interactive function skipped.")
print("-" * 50)

# Problem 50: Find all indices of a target element in a list using a loop with continue.
print("Problem 50: Find all indices of a target element in a list using a loop with continue.")
lst = [1, 2, 3, 2, 4, 2]
target = 2
indices = []
for index, value in enumerate(lst):
   if value != target:
      continue
   indices.append(index)
print(indices)
print("-" * 50)

# Problem 51: Use a loop with 'else' to confirm if all elements in a list are unique.
print("Problem 51: Use a loop with 'else' to confirm if all elements in a list are unique.")
lst = [1, 2, 3, 4]
seen = set()
for item in lst:
```

```
    if item in seen:
        print("Not all elements are unique")
        break
    seen.add(item)
else:
    print("All elements are unique")

lst = [1, 2, 2, 3]
seen = set()
for item in lst:
    if item in seen:
        print("Not all elements are unique")
        break
    seen.add(item)
else:
    print("All elements are unique")
print("-" * 50)

# Problem 52: Implement a password checker that allows three attempts using loops and break.
print("Problem 52: Implement a password checker simulation.")
# Uncomment the lines below to run the interactive function
# correct_password = "secret"
# attempts = 3
# while attempts > 0:
#     pwd = input("Enter password: ")
#     if pwd == correct_password:
#         print("Access granted")
#         break
#     else:
#         attempts -= 1
#         print(f"Incorrect password. Attempts left: {attempts}")
```

```
# else:
#     print("Access denied")
print("Interactive function skipped.")
print("-" * 50)

# Problem 53: Use 'pass' in a loop to create a placeholder for future code.
print("Problem 53: Use 'pass' in a loop to create a placeholder for future code.")
for i in range(5):
    if i == 2:
        pass  # Future implementation
    print(i)
print("-" * 50)

# Problem 54: Combine multiple loop controls in a single loop.
print("Problem 54: Combine multiple loop controls in a single loop.")
for i in range(1, 10):
    if i == 3:
        continue
    if i == 7:
        break
    if i == 5:
        pass
    print(i)
else:
    print("Loop completed without break")
print("-" * 50)

================================================================================
End of Worksheet
================================================================================
```