Certainly! Here are 20 problems along with their solutions based on the topics of Python's basic syntax, string manipulation, numeric data types, conversion functions, and the `print()` function:

Problems:

1. **String Concatenation:**
   - Concatenate two strings and print the result.

2. **Numeric Operations:**
   - Perform addition, subtraction, multiplication, and division of two numbers and print the results.

3. **String Indexing:**
   - Given a string, print the first and last characters using indexing.

4. **Conversion Functions:**
   - Take a numeric input from the user and print its square using the `int()` and `print()` functions.

5. **Formatted String:**
   - Create a formatted string that includes your name, age, and a brief introduction.

6. **String Slicing:**
   - Given a string, extract a substring using string slicing and print it.

7. **Type Checking:**
   - Check the type of a variable and print the result.

8. **String Repetition:**
   - Take a string input from the user and print it repeated three times.

9. **List Conversion:**
   - Given a string, convert it to a list of characters and print the list.

10. **Boolean Conversion:**
    - Take an integer input from the user and print whether it is even or odd using boolean conversion.

11. **Formatting with `%`:**
    - Create a formatted string using the `%` method to display the current date and time.

12. **Float Conversion:**
    - Take a float input from the user, convert it to an integer, and print the result.

13. **Random Number Generation:**
    - Generate a random integer between 1 and 10 using the `random` module and print it.

14. **Decimal Arithmetic:**
    - Take two decimal inputs from the user, add them, and print the result using the `decimal` module.

15. **Exponentiation:**
    - Calculate and print the value of 2 to the power of 5.

16. **String Membership:**
    - Check if a given substring is present in a string and print the result.

17. **String Formatting with f-strings:**
    - Use an f-string to create a formatted string with your favorite color and print it.

18. **Conversion Error Handling:**
    - Handle the exception that occurs when trying to convert a non-numeric string to an integer and print a custom error message.

19. **Hexadecimal Conversion:**
    - Convert the decimal number 255 to a hexadecimal string and print it.

20. **Printing Patterns:**
    - Use the `print()` function to create a pattern of asterisks, forming a triangle or rectangle.

Solutions:

Apologies for the confusion. Let me provide answers to the first few problems as examples. Feel free to ask for solutions to specific problems or additional explanations for any of them.

Sample Solutions:

1. **String Concatenation:**
   ```python
   str1 = "Hello"
   str2 = "World"
   result = str1 + " " + str2
   print(result)
   ```

2. **Numeric Operations:**
   ```python
   num1 = 10
   num2 = 5
   print("Sum:", num1 + num2)
   print("Difference:", num1 - num2)
   print("Product:", num1 * num2)
   print("Quotient:", num1 / num2)
   ```

3. **String Indexing:**
   ```python
   my_string = "Python"
   first_char = my_string[0]
   last_char = my_string[-1]
   print("First Character:", first_char)
   print("Last Character:", last_char)
   ```

```

4. **Conversion Functions:**
   ```python
   user_input = input("Enter a number: ")
   num = int(user_input)
   square = num ** 2
   print("Square:", square)
   ```

5. **Formatted String:**
   ```python
   name = "Alice"
   age = 30
   introduction = "My name is %s and I am %d years old." % (name, age)
   print(introduction)
   ```

6. **String Slicing:**
   ```python
   my_string = "Hello, World!"
   substring = my_string[7:12]
   print(substring)
   ```

7. **Type Checking:**
   ```python
   my_variable = 42
   print("Type of my_variable:", type(my_variable))
   ```

8. **String Repetition:**
   ```python
   user_input = input("Enter a string: ")
   repeated_string = user_input * 3
   print("Repeated String:", repeated_string)
   ```

9. **List Conversion:**
   ```python
   my_string = "Python"
   char_list = list(my_string)
   print("List of Characters:", char_list)
   ```

10. **Boolean Conversion:**
    ```python

```
user_input = int(input("Enter an integer: "))
is_even = user_input % 2 == 0
print("Is it even?", is_even)
```

11. **Formatting with `%`:**
```python
import datetime
current_datetime = datetime.datetime.now()
formatted_datetime = "Current date and time: %s" % current_datetime
print(formatted_datetime)
```

12. **Float Conversion:**
```python
float_input = float(input("Enter a float: "))
int_result = int(float_input)
print("Converted to Integer:", int_result)
```

13. **Random Number Generation:**
```python
import random
random_number = random.randint(1, 10)
print("Random Number:", random_number)
```

14. **Decimal Arithmetic:**
```python
from decimal import Decimal
decimal1 = Decimal('2.5')
decimal2 = Decimal('3.7')
result = decimal1 + decimal2
print("Decimal Result:", result)
```

15. **Exponentiation:**
```python
exponent_result = 2 ** 5
print("2 to the power of 5:", exponent_result)
```

16. **String Membership:**
```python
my_string = "Python"
is_present = "Py" in my_string
print("Is 'Py' present?", is_present)
```

17. **String Formatting with f-strings:**
    ```python
    favorite_color = "Blue"
    formatted_string = f"My favorite color is {favorite_color}."
    print(formatted_string)
    ```

18. **Conversion Error Handling:**
    ```python
    user_input = input("Enter a number: ")
    try:
        num = int(user_input)
        print("Conversion successful:", num)
    except ValueError:
        print("Invalid input. Please enter a valid number.")
    ```

19. **Hexadecimal Conversion:**
    ```python
    decimal_number = 255
    hex_string = hex(decimal_number)
    print("Hexadecimal:", hex_string)
    ```

20. **Printing Patterns:**
    ```python
    for i in range(5):
        print("*" * (i + 1))
    ```