1. Check if "cat" and "hat" appear the same number of times

- str_input = input("Enter a string with 'cat' and 'hat': ")

```
count_cat = str_input.count("cat")
count_hat = str_input.count("hat")
if count_cat == count_hat:
    print(True)
else:
    print(False)
```

2. Print sum of squares pattern

```
n = int(input("Enter a natural number n: "))
for i in range(1, n+1):
    sum_squares = 0
    for j in range(1, i+1):
        sum_squares += j * j
    print(sum_squares)
```

3. Sum of N natural numbers using while loop

```
n = int(input("Enter a natural number: "))
sum_n = 0
while n > 0:
    sum_n += n
    n -= 1
print("The sum of natural numbers is:", sum_n)
```

4. Remove duplicate values across dictionary values

```
my_dict = {'a': [1, 2, 2], 'b': [2, 3, 3], 'c': [1, 4, 4]}
new_dict = {}
for key in my_dict:
    unique_values = []
    for value in my_dict[key]:
        if value not in unique_values:
            unique_values.append(value)
    new_dict[key] = unique_values
```

```
print("Dictionary without duplicates:", new_dict)
```

5. Count character frequency in a string

```
s = input("Enter a string: ")
freq = {}
for ch in s:
    if ch in freq:
        freq[ch] += 1
    else:
        freq[ch] = 1
print("Character frequencies:", freq)
```

6. Find dictionary value if key exists in list and dictionary

```
my_list = [1, 2, 3, 4, 5]
my_dict = {3: "apple", 5: "banana", 7: "cherry"}
K = int(input("Enter a key number: "))
if K in my_list and K in my_dict:
    print("Value from dictionary:", my_dict[K])
else:
    print("Key not found in both list and dictionary.")
```

7. Check if a number is a power of 2

```
n = int(input("Enter a number: "))
num = n
if num <= 0:
    print("Not a power of 2")
else:
    while num != 1:
        if num % 2 != 0:
            print("Not a power of 2")
            break
        num = num // 2
    else:
        print("Is a power of 2")
```

8. Print coordinates with even x and y

```
coordinates = [(2, 4), (1, 3), (6, 8), (7, 2), (0, 0)]
result = []
for x, y in coordinates:
    if x % 2 == 0 and y % 2 == 0:
        result.append((x, y))
print("Coordinates with even x and y:", result)
```

9. Check if all elements in a list are unique

```
my_list = input("Enter numbers separated by spaces: ").split()
if len(set(my_list)) == len(my_list):
    print("All elements are unique.")
else:
    print("Some elements are repeated.")
```

10. Find missing element in two arrays

```
arr1 = list(map(int, input("Enter numbers for array 1 (separated by spaces): ").split()))
print("Array 1:", arr1)
arr2 = list(map(int, input("Enter numbers for array 2 (separated by spaces): ").split()))
print("Array 2:", arr2)
missing = list(set(arr1) ^ set(arr2))
print("Missing element:", missing[0] if missing else "No missing element")
```

11. Swap elements inside each tuple

```
tuples = [(1, 2), (3, 4), (5, 6)]
swap = []
for a, b in tuples:
    swap.append((b, a))
print("Tuples after swapping:", swap)
```

12. Calculate column-wise average of tuple of tuples

```
tup = ((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3, 4))
n = len(tup[0])
averages = []
for col in range(n):
```

```
col_sum = 0
for row in tup:
col_sum += row[col]
avg = col_sum / len(tup)
averages.append(avg)
print("Column averages:", averages)
```

NumPY

1. Reverse NumPy array of even numbers

```
import numpy as np
arr = np.arange(2, 101, 2)
arr = np.flip(arr)
```

2. 8x8 checkerboard pattern

```
import numpy as np
checkerboard = np.zeros((8, 8), dtype=int)
checkerboard[::2, ::2] = 1
checkerboard[1::2, 1::2] = 1
print(checkerboard)
```

3. 5x5 identity matrix with diagonal 1 to 5

```
import numpy as np
matrix = np.zeros((5, 5))
np.fill_diagonal(matrix, [1, 2, 3, 4, 5])
print(matrix)
```

4. Extract border elements from 5x5 array

```
import numpy as np
arr = np.random.rand(5, 5)
border = []
border.extend(arr[0])  # First row
border.extend(arr[-1])  # Last row
for i in range(1, 4):
 border.append(arr[i][0])  # First column (excluding corners)
 border.append(arr[i][-1])  # Last column (excluding corners)
```

```
print("Border elements:", border)
```

5. Extract unique rows from 2D array

```
import numpy as np
a = np.array([[1, 2], [3, 4], [1, 2]])
unique = np.unique(a, axis=0)
print("Unique rows:", unique)
```

7. Calculate determinant of 3x3 matrix

```
import numpy as np
m = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
det = (m[0, 0] * (m[1, 1] * m[2, 2] - m[1, 2] * m[2, 1]) -
      m[0, 1] * (m[1, 0] * m[2, 2] - m[1, 2] * m[2, 0]) +
      m[0, 2] * (m[1, 0] * m[2, 1] - m[1, 1] * m[2, 0]))
print("Determinant:", det)
```

8. Append to an empty NumPy array

```
import numpy as np
arr = np.array([])
arr = np.append(arr, [1, 2, 3])
print(arr)
```

9. Compare np.append vs np.concatenate

```
import numpy as np
import time
data = [1]
start = time.time()
a = np.array([])
for _ in range(10000):
   a = np.append(a, data)
print("Time for append:", time.time() - start)
start = time.time()
b = np.array([])
for_in range(10000):
   b = np.append(b, data)
```

```
print("Time for append (again):", time.time() - start)
```

10. Append even numbers to array

```
import numpy as np
arr = np.array([1, 2, 3])
new_nums = [4, 5, 6]
for x in new_nums:
    if x % 2 == 0:
        arr = np.append(arr, x)
print(arr)
```

11. Generate 200x300 grayscale image

```
import numpy as np
image = np.full((200, 300), 128)
print("Grayscale image shape:", image.shape)
print("Sample pixel value:", image[0, 0])
```

12. Extract 100x100 region from 500x600 image

Practical Questions

11. Convert 3.14 to integer

```
num = 3.14
print(int(num))
# 3
```

12. Output of print(type(True))

```
print(type(True))
 # <class 'bool'>
```

13. Convert "123" to integer and add 10

```
a = int("123")
b = a + 10
print(b)
# 133
```

14. Output of print(bool(0), bool("Hello"), bool(None))

```
print(bool(0), bool("Hello"), bool(None))
# False True False
```

15. Why does print(10 + "20") give an error? Fix it.

Question: Explain the error and fix it.

It errors because you can't add an integer (`10`) and a string (`"20"`).

Fix:

```
print(10 + int("20"))
 # 30
```

16. Output of print(int(5.6))

```
print(int(5.6))
# 5
```

17. Check if var = "Python" is of type str

```
var = "Python"
print(type(var))
  # <class 'str'>
```

18. Result of float("3.14") + 1

```
print(float("3.14") + 1)
 # 4.14
```

19. Output of print(str(10) + str(20))

```
print(str(10) + str(20))
  # 1020
```

20. Difference between "5" and 5

- "5" is a string (text).
- 5 is an integer (number).

List and Strings Questions

3. Convert list of tuples to dictionary

```
def convert_to_dict(lst):
    result = {}
    for key, value in lst:
        if key in result:
            result[key].append(value)
```

```
        else:
            result[key] = [value]
    return result
# Test
lst = [('a', 1), ('b', 2), ('a', 3)]
print(convert_to_dict(lst))
 # {'a': [1, 3], 'b': [2]}
```

6. Intersection of two lists without duplicates

```
def list_intersection(lst1, lst2):
    result = []
    for x in lst1:
        if x in lst2 and x not in result:
            result.append(x)
    return result
# Test
lst1 = [1, 2, 3, 2, 4]
lst2 = [2, 2, 4, 5]
print(list_intersection(lst1, lst2))
  # [2, 4]
```

10. Find missing number in list

Question: Given a list of n-1 integers from 1 to n, find the missing number. Example: [1, 2, 4, 5] → 3.

python

Copy

```
def find_missing_number(lst):
    n = len(lst) + 1
    expected_sum = n * (n + 1) // 2
    actual_sum = sum(lst)
    return expected_sum - actual_sum
# Test
lst = [1, 2, 4, 5]
print(find_missing_number(lst))  #3
```