

Question#1

1. **5**

- Data type: int

2. **5.0**

- Data type: float

3. **5 > 1**

- Data type: bool

4. **'5'**

- Data type: str

5. **5 * 2**

- Data type: int

6. **'5' * 2**

- Data type: str

7. **'5' + '2'**

- Data type: str

8. **5 / 2**

- Data type: float

9. **5 % 2**

- Data type: int

10. **{5, 2, 1}**

- Data type: set

11. **5 == 3**

- Data type: bool

12. **Pi (the number)**

- Data type: float

Question#2

a. Number of letters in 'Supercalifragilisticexpialidocious'

```
word = 'Supercalifragilisticexpialidocious'
```

```
letters_count = len(word)
```

b. Check if 'ice' is a substring in 'Supercalifragilisticexpialidocious'

```
contains_ice = 'ice' in word
```

c. Find the longest word

```
words = ['Supercalifragilisticexpialidocious', 'Honorificabilitudinitatibus',  
         'Bababadalgharaghtakamminarronnkonn']
```

```
longest_word = max(words, key=len)
```

d. Find the first and last composer in alphabetical order

```
composers = ['Berlioz', 'Borodin', 'Brian', 'Bartok', 'Bellini', 'Buxtehude', 'Bernstein']
```

```
first_composer = min(composers)
```

```
last_composer = max(composers)
```

```
letters_count, contains_ice, longest_word, first_composer, last_composer
```

Question#3

```
import math
```

```
def triangleArea(a, b, c):
```

```
    # Calculate semi-perimeter
```

```
    s = (a + b + c) / 2
```

```
    # Calculate the area using Heron's formula
```

```
    area = math.sqrt(s * (s - a) * (s - b) * (s - c))
```

```
    return area
```

```
# Test the function
```

```
triangleArea(2, 2, 2)
```

Question#4

Function to separate odd and even integers into separate arrays

```
def separate_odd_even(numbers):
```

```
    even_numbers = []
```

```
    odd_numbers = []
```

```
    for num in numbers:
```

```
        if num % 2 == 0:
```

```
            even_numbers.append(num)
```

```
        else:
```

```
            odd_numbers.append(num)
```

```
    return even_numbers, odd_numbers
```

Input: Number of elements and the elements themselves

```
n = int(input("Input the number of elements to be stored in the array: "))
```

```
numbers = []
```

```
for i in range(n):
```

```
    element = int(input(f"element - {i}: "))
```

```
    numbers.append(element)
```

Separate the numbers into odd and even

```
even_numbers, odd_numbers = separate_odd_even(numbers)
```

Output the results

```
print("The Even elements are:")
```

```
for num in even_numbers:
```

```
    print(num, end=' ')
```

```
print("\n\nThe Odd elements are:")
```

```
for num in odd_numbers:
```

```
    print(num, end=' ')
```

Question#5

Part a: Function to check if a point (x, y) is inside a rectangle

```
def inside(x, y, x1, y1, x2, y2):  
    return x1 <= x <= x2 and y1 <= y <= y2
```

Testing part a

```
test1 = inside(1, 1, 0, 0, 2, 3) # Expected: True  
test2 = inside(-1, -1, 0, 0, 2, 3) # Expected: False
```

Part b: Checking if the point (1,1) lies in both rectangles

```
test_both = inside(1, 1, 0.3, 0.5, 1.1, 0.7) and inside(1, 1, 0.5, 0.2, 1.1, 2)
```

```
test1, test2, test_both
```

Question#6

```
def pig(word):  
    word = word.lower() # Convert the word to lower case  
    vowels = 'aeiou'
```

```
    if word[0] in vowels:  
        return word + 'way'  
    else:  
        return word[1:] + word[0] + 'ay'
```

Testing the function

```
print(pig('happy')) # Expected: 'appyhay'  
print(pig('Enter')) # Expected: 'enterway'
```

Question#7

Reading the content of the file

file_path = '/mnt/data/bloodtype1.txt'

def bldcount(file_path):

Reading the blood types from the file

with open(file_path, 'r') as file:

blood_types = file.read().split()

Define possible blood types

blood_type_counts = {'A': 0, 'B': 0, 'AB': 0, 'O': 0, 'OO': 0}

Count occurrences of each blood type

for blood_type in blood_types:

if blood_type in blood_type_counts:

blood_type_counts[blood_type] += 1

Print the results

for blood_type, count in blood_type_counts.items():

if count == 0:

print(f"There are no patients of blood type {blood_type}.")

elif count == 1:

print(f"There is one patient of blood type {blood_type}.")

else:

print(f"There are {count} patients of blood type {blood_type}.")

Call the function with the provided file path

bldcount(file_path)

Question#8

Define the function to convert currency based on the given rates

```
def curconv(currency, amount):
```

```
    # Read the file with currency rates
```

```
    rates = {}
```

```
    with open('/mnt/data/currencies.txt', 'r') as file:
```

```
        for line in file:
```

```
            parts = line.split()
```

```
            rates[parts[0]] = float(parts[1])
```

```
    # Convert the amount to USD using the rate
```

```
    if currency in rates:
```

```
        return amount * rates[currency]
```

```
    else:
```

```
        return "Currency not found."
```

```
# Testing the function
```

```
print(curconv('EUR', 100)) # Expected: 122.96544
```

```
print(curconv('JPY', 100)) # Expected: 1.241401
```

Question#9

- **Trying to add incompatible variables, as in `6 + 'a'`:**
 - **Exception:** `TypeError`
 - **Reason:** You cannot add an integer (6) to a string ('a'), as they are incompatible types.
- **Referring to the 12th item of a list that has only 10 items:**
 - **Exception:** `IndexError`
 - **Reason:** This happens because you are trying to access an index that is out of range for the list.

- **Using a value that is out of range for a function's input, such as calling `math.sqrt(-1.0)`:**
 - **Exception:** `ValueError`
 - **Reason:** The `math.sqrt()` function expects a non-negative number, and passing a negative number like `-1.0` is outside the valid range.
- **Using an undeclared variable, such as `print(x)` when `x` has not been defined:**
 - **Exception:** `NameError`
 - **Reason:** The variable `x` has not been defined, so trying to reference it will raise a `NameError`.
- **Trying to open a file that does not exist, such as mistyping the file name or looking in the wrong directory:**
 - **Exception:** `FileNotFoundError`
 - **Reason:** This occurs when you attempt to open a file that the program cannot locate or does not exist in the specified directory.

Question#10

```
def frequencies(text):
    # Define the string of letters to track
    letters = 'abcdefghijklmnopqrstuvwxyz'

    # Create a list of zeros with length equal to the number of letters
    freq_list = [0] * len(letters)

    # Iterate through each character in the input text
    for char in text.lower():
        if char in letters:
            index = letters.index(char)
            freq_list[index] += 1

    return freq_list
```