```
#1) Pass a tuple to divmod() and obtain the quotient and the remainder.
num_tuple = (17, 5)
quotient, remainder = divmod(num_tuple[0], num_tuple[1])

# Output the result
print(f"Quotient: {quotient}, Remainder: {remainder}")
```

```
Quotient: 3, Remainder: 2
```

```
#2) Unpack the tuple in 5 variables, each holding 1 value.
# Define a tuple with 5 values
my_tuple = (10, 20, 30, 40, 50)

# Unpack the tuple into 5 variables
a, b, c, d, e = my_tuple

# Output the values of the variables
print(f"a = {a}, b = {b}, c = {c}, d = {d}, e = {e}")
```

```
a = 10, b = 20, c = 30, d = 40, e = 50
```

```
#3) Unpack the tuple in 5 variables such that 1st value gets stored in 1st variable  x a
#value in y and all other values in between into disposable variables.

# Define a tuple with 5 values
my_tuple = (10, 20, 30, 40, 50)

# Unpack the tuple, assigning the first and last values to x and y, and the rest to a di
x, *_, y = my_tuple

# Output the values of x and y
print(f"x = {x}, y = {y}")
```

```
x = 10, y = 50
```

```
#1. A list contains names of boys and girls as its elements. Boys’ names are stored as t
#number of boys and girls in the list. (Hint: use isinstance(ele,tuple))
# Define a list containing boys' names as tuples and girls' names as strings
names_list = [("John", "Tom"), "Alice", "Sara", ("Peter", "Sam"), "Eva", ("Jack",)]

# Initialize counters
boys_count = 0
girls_count = 0

# Loop through the list to count boys and girls
for ele in names_list:
    if isinstance(ele, tuple):  # Check if it's a tuple (boys' names)
        boys_count += 1
    else:  # Else it's a girl's name (string)
        girls_count += 1

# Output the results
print(f"Number of boys: {boys_count}")
print(f"Number of girls: {girls_count}")
```

```
Number of boys: 3
Number of girls: 3
```

```
#2. A list contains tuples containing roll no., name and age of student. Write a python
#program to create three lists separately for roll no., name and age
# Define a list of tuples with roll no., name, and age
students_list = [(1, "John", 18), (2, "Alice", 19), (3, "Bob", 20), (4, "Sara", 18)]

# Initialize empty lists for roll numbers, names, and ages
roll_numbers = []
names = []
ages = []

# Loop through the list of students and unpack the values into the respective lists
for student in students_list:
    roll_no, name, age = student
    roll_numbers.append(roll_no)
    names.append(name)
    ages.append(age)

# Output the three lists
print("Roll Numbers:", roll_numbers)
print("Names:", names)
print("Ages:", ages)
```

```
Roll Numbers: [1, 2, 3, 4]
Names: ['John', 'Alice', 'Bob', 'Sara']
Ages: [18, 19, 20, 18]
```

```
#3. Suppose a date is represented as a tuple (d, m, y).
#Create two date tuples and find the number of days between the two dates.
from datetime import date

# Define the two date tuples
date1_tuple = (15, 3, 2022)  # 15 March 2022
date2_tuple = (25, 4, 2023)  # 25 April 2023

# Convert the tuples into date objects (note: date(year, month, day))
date1 = date(date1_tuple[2], date1_tuple[1], date1_tuple[0])
date2 = date(date2_tuple[2], date2_tuple[1], date2_tuple[0])

# Calculate the difference
difference = abs((date2 - date1).days)

# Print the result
print(f"Number of days between {date1_tuple} and {date2_tuple}: {difference} days")
```

```
Number of days between (15, 3, 2022) and (25, 4, 2023): 406 days
```

```
#4. Create a list of tuples containing a food item and its price.
#Sort the tuples in descending order by price.
# List of food items and their prices
menu = [("Burger", 5.99), ("Pizza", 8.50), ("Salad", 4.25), ("Pasta", 7.30), ("Fries", 2
# Sort the list by price in descending order
menu_sorted = sorted(menu, key=lambda item: item[1], reverse=True)

# Print the sorted menu
print("Menu sorted by price (high to low):")
for food, price in menu_sorted:
    print(f"{food}: ${price:.2f}")
```

```
Menu sorted by price (high to low):
Pizza: $8.50
Pasta: $7.30
Burger: $5.99
Salad: $4.25
Fries: $2.99
```

#5. Remove empty tuple(s) from the list of tuples.

```
# List containing some empty and non-empty tuples
tuples_list = [(), (1, 2), (), (3,), (4, 5, 6), (), ("a",)]

# Remove empty tuples
filtered_list = [t for t in tuples_list if t]

# Print the result
print("List after removing empty tuples:")
print(filtered_list)
```

```
List after removing empty tuples:
[(1, 2), (3,), (4, 5, 6), ('a',)]
```

#6. Modify an element of a tuple.

```
# Original tuple
my_tuple = (10, 20, 30, 40)

# Convert to list
temp_list = list(my_tuple)

# Modify an element (e.g., change 30 to 300)
temp_list[2] = 300

# Convert back to tuple
my_tuple = tuple(temp_list)

# Print the modified tuple
print("Modified tuple:", my_tuple)
```

```
Modified tuple: (10, 20, 300, 40)
```

#7. Delete an element of a tuple.

```
# Original tuple
my_tuple = (10, 20, 30, 40)

# Convert to list
temp_list = list(my_tuple)
```

```
# Delete the element at index 1 (value 20)
del temp_list[1]

# Convert back to tuple
my_tuple = tuple(temp_list)

# Print the modified tuple
print("Tuple after deleting an element:", my_tuple)
```

```
Tuple after deleting an element: (10, 30, 40)
```