BÀI TẬP TUPLE - CÓ LỜI GIẢI CHI TIẾT

BÀI 1: DANH SÁCH BẠN BÈ
friends = [("Bob", "Male"), ("Anna", "Female"), ("Max", "Male")]

a) Cho biết chiều dài của friends
=> Lời giải:
len(friends) trả về 3 vì có 3 phần tử trong list.

b) Lấy ra phần tử đầu, cuối và giữa và kiểm tra kiểu dữ liệu
=> Lời giải (code):

```
first = friends[0]
middle = friends[len(friends)//2]
last = friends[-1]

print(first)   # ("Bob", "Male")
print(middle)  # ("Anna", "Female")
print(last)    # ("Max", "Male")
print(type(first))  # <class 'tuple'>
```

Giải thích: Truy cập bằng chỉ số; mỗi phần tử là một tuple nên type(...) trả về tuple.

c) Nhập vào name và gender, sau đó append vào friends
=> Lời giải (code):

```
name = input("Name: ")
gender = input("Gender: ")
friends.append((name, gender))
print(friends)
```

Giải thích: append nhận một phần tử mới; vì mỗi phần tử friends là tuple có 2 giá trị, ta append((name, gender)).

BÀI 2: TÍNH TỔNG VÀ TRUNG BÌNH
numbers = (5, 10, 15, 20, 25)

a) Tính tổng của các phần tử
=> Lời giải:
total = sum(numbers) # 75

b) Tính giá trị trung bình
=> Lời giải:
average = sum(numbers) / len(numbers) # 75 / 5 = 15.0

c) In ra phần tử lớn nhất và nhỏ nhất
=> Lời giải:
largest = max(numbers) # 25
smallest = min(numbers) # 5

Giải thích: sum, len, max, min là các hàm built-in dùng trực tiếp trên tuple.

BÀI 3: ĐẾM PHẦN TỬ TRÙNG
colors = ("red", "blue", "green", "red", "red", "blue")

a) Đếm số lần xuất hiện của "red"
=> Lời giải:
count_red = colors.count("red") # 3

b) Tìm vị trí đầu tiên của "blue"
=> Lời giải:
index_blue = colors.index("blue") # 1 (vị trí bắt đầu từ 0)

Giải thích: tuple hỗ trợ count() và index(). index() ném lỗi nếu không tìm thấy, nên có thể dùng in kiểm tra trước.

BÀI 4: GỘP HAI TUPLE
tuple1 = (1, 2, 3)
tuple2 = (4, 5, 6)

a) Gộp lại thành một tuple mới
=> Lời giải:
combined = tuple1 + tuple2 # (1, 2, 3, 4, 5, 6)

b) In ra tuple mới
print(combined)

Giải thích: Toán tử + nối hai tuple, kết quả vẫn là tuple.

BÀI 5: CHUYỂN LIST ↔ TUPLE
fruits_list = ["apple", "banana", "cherry"]

a) Chuyển list thành tuple
=> Lời giải:
fruits_tuple = tuple(fruits_list) # ("apple", "banana", "cherry")

b) Chuyển tuple ngược lại thành list
=> Lời giải:
fruits_list_again = list(fruits_tuple) # ["apple", "banana", "cherry"]

Giải thích: tuple(...) và list(...) để chuyển đổi giữa hai kiểu.

BÀI 6: TUPLE LỒNG NHAU
students = (
    ("Alice", (9, 8, 10)),
    ("Bob", (7, 6, 9)),
    ("Anna", (10, 10, 9))
)

a) In điểm trung bình của từng học sinh
=> Lời giải (code):
for name, scores in students:
    avg = sum(scores) / len(scores)
    print(f"{name}: {avg:.2f}")

```
# Kết quả:
# Alice: 9.00
# Bob: 7.33
# Anna: 9.67

b) In ra học sinh có điểm trung bình cao nhất
=> Lời giải (code):
best_name = None
best_avg = -1

for name, scores in students:
    avg = sum(scores) / len(scores)
    if avg > best_avg:
        best_avg = avg
        best_name = name

print(f"Học sinh có điểm trung bình cao nhất: {best_name} ({best_avg:.2f})")
# Kết quả: Anna (9.67)

Giải thích: Duyệt qua từng tuple, tính trung bình, so sánh để tìm max.

----------------------------------------

BÀI 7: QUẢN LÝ SẢN PHẨM (Nhập và lưu tuple)
Yêu cầu: Nhập tên, giá, số lượng; lưu thành tuple; thêm vào list products; in danh sách.

=> Lời giải (code):
products = []

name = input("Tên sản phẩm: ")
price = float(input("Giá: "))
qty = int(input("Số lượng: "))
product = (name, price, qty)
products.append(product)

print("Danh sách sản phẩm:")
for p in products:
```

```
print(p)
```

Giải thích: Dùng tuple để lưu từng sản phẩm (không sửa đổi sau khi tạo). Nếu muốn sửa, có thể chuyển tuple sang list rồi chỉnh rồi chuyển lại.

PHỤ LỤC: GỢI Ý BÀI TẬP MỞ RỘNG

1) Viết hàm nhận list của tuple (name, age) và trả về tên người lớn tuổi nhất.

2) Viết hàm nhóm các tuple theo giới tính (ví dụ bài friends), trả về dict {'Male': [...], 'Female': [...]}.

3) Viết hàm nhận tuple số nguyên và trả về tuple đảo ngược, không dùng slicing.