I. **Phân biệt toán tử định dạng chuỗi và hàm định dạng chuỗi có sẵn trong**

**gói thư viện chuẩn Python? Cho năm ví dụ minh họa tương ứng?**

**1. Toán tử định dạng chuỗi (%)**

Toán tử % được sử dụng để định dạng chuỗi, giống như cách thức hoạt động trong ngôn ngữ C.

Vd:a=18

Tuoi=”toi nam nay %d tuoi” %a

Print(Tuoi)

Output: toi nam nay 18 tuoi

Vd2: pi = 3.14159265359

Bi="Giá trị của Pi là khoảng %.3f." % pi

print(Bi)

Output: "Giá trị của Pi là khoảng 3.141."

Vd3: name = "John"

ten = "Tên tôi là %s." % name

print(Ten)

# Output: "Tên tôi là John."

Vd4:

name = "Phong" age = 19

Tentuoi= "Tôi tên là %s và tôi %d tuổi." % (name, age)

print(Tentuoi)

# Output: "Tôi tên là Phongvà tôi 19 tuổi."

Vd5:

percentage = 75

formatted\_string = "Bạn đã hoàn thành %d%% công việc." % percentage print(formatted\_string)

# Output: "Bạn đã hoàn thành 75% công việc."

**2. Hàm định dạng chuỗi (str.format())**

Hàm str.format() là một cách linh hoạt và mạnh mẽ hơn để định dạng chuỗi.

Vd:a=18

Tuoi=”toi nam nay {} tuoi” format(a)

Print(Tuoi)

Output: “toi nam nay 18 tuoi”

# Ví dụ 2: Định dạng số thực

pi = 3.14159265359

formatted\_string = "Giá trị của Pi là khoảng {:.2f}.".format(pi)

print(formatted\_string)

# Output: "Giá trị của Pi là khoảng 3.14."

# Ví dụ 3: Định dạng chuỗi

name = "John"

formatted\_string = "Tên tôi là {}.".format(name)

print(formatted\_string)

# Output: "Tên tôi là John."

# Ví dụ 4: Định dạng nhiều giá trị

name = "Alice" age = 30

formatted\_string = "Tôi tên là {} và tôi {} tuổi.".format(name, age)

print(formatted\_string)

# Output: "Tôi tên là Alice và tôi 30 tuổi."

# Ví dụ 5: Định dạng chỉ số phần trăm

percentage = 75

formatted\_string = "Bạn đã hoàn thành {}% công việc.".format(percentage)

print(formatted\_string)

# Output: "Bạn đã hoàn thành 75% công việc."

II**. Viết chương trình xuất ra số ngẫu nhiên trong một đoạn bất kỳ t cho**

**trước?**

start = int(input("Nhập số bắt đầu của đoạn: "))

end = int(input("Nhập số kết thúc của đoạn: "))

random\_number = random.randint(start, end)

print(f"Số ngẫu nhiên trong đoạn [{start}, {end}] là: {random\_number}")

**III. Khác biệt cơ bản giữa list và tuple?**

-List:Là kiểu dữ liệu có thể thay đổi

-Tuple:Là kiểu dữ liệu không thể thay đổi

**IV. Ứng dụng kiểu dữ liệu tuple trong thực tế?**

1. Lưu trữ dữ liệu cố định hoặc không thay đổi
2. Sử dụng làm khóa trong từ điển
3. Truyền nhiều giá trị trong các hàm
4. Làm dữ liệu phức tạp đơn giản hơn
5. Bảo vệ dữ liệu quan trọng

Tuple là một công cụ mạnh mẽ trong Python nhờ tính bất biến của nó, giúp bảo vệ dữ liệu, sử dụng làm khóa trong từ điển, và truyền dữ liệu phức tạp giữa các hàm. Trong thực tế, tuple được sử dụng nhiều để lưu trữ thông tin cố định, gom nhóm các giá trị và bảo vệ dữ liệu quan trọng khỏi những thay đổi không mong muốn.