

سؤال ۱: پاکسازی و قالب‌بندی رشته

یک متغیر به شکل زیر تعریف شده است. با استفاده از متدهای رشته، آن را به یک رشته تمیز و استاندارد تبدیل کنید به طوری که:

۱. تمام حروف به حروف کوچک تبدیل شوند.

۲. فاصله‌های اضافی در ابتدا و انتهای رشته حذف شوند.

۳. در نهایت، نام را در پیام "welcome, {full_name}!" با استفاده از f-string چاپ کنید.

```
full_name: str = " \tHadi Farhadi "  
  
# کد خود را اینجا بنویسید  
# ...  
  
# خروجی مورد انتظار:  
# welcome, hadi farhadi!
```

سؤال ۲: شمارش تکرار کاراکترها در رشته

کدی بنویسید که یک رشته ورودی را از کاربر دریافت کرده و تعداد تکرار هر کاراکتر را در قالب یک دیکشنری محاسبه و چاپ کند. (راهنمایی: از متد `get()` دیکشنری استفاده کنید)

```
input_str = input("Enter a string: ") # مثلا ورودی: "hello"  
letters_count = {}  
  
# کد خود را اینجا بنویسید  
# ...  
  
print(letters_count)  
  
# "hello": خروجی مورد انتظار برای ورودی  
# {'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

سؤال ۳: استخراج داده از لیست تودرتو

با توجه به لیست تودرتوی gold_data که شامل قیمت‌های طلا (Open, Close, Low, High) است، یک لیست جدید به نام high_prices با استفاده از **List Comprehension** بسازید که فقط شامل قیمت‌های High (عنصر چهارم هر لیست داخلی) باشد.

```
gold_data = [  
    [1954.32, 1956.78, 1953.15, 1957.24],  
    [1956.78, 1958.45, 1955.62, 1959.83],  
    [1958.45, 1957.23, 1956.11, 1959.02],  
    [1957.23, 1955.67, 1954.89, 1958.15]  
]  
  
high_prices = # کد خود را اینجا بنویسید  
  
print(high_prices)  
  
# خروجی مورد انتظار:  
# [1957.24, 1959.83, 1959.02, 1958.15]
```

سؤال ۴: تجمیع لیست‌ها در دیکشنری

سه لیست زیر داده شده است. با استفاده از تابع zip و List Comprehension، لیستی از دیکشنری‌ها بسازید که هر دیکشنری اطلاعات یک محصول را شامل شود.

```
products = ["Laptop", "Mouse", "Keyboard"]  
prices = [1500, 25, 75]  
quantities = [10, 100, 50]  
  
# کد شما اینجا  
# inventory_list = ...  
  
print(inventory_list)  
  
# خروجی مورد انتظار:  
# [{'name': 'Laptop', 'price': 1500, 'quantity': 10}, {'name': 'Mouse', 'price': 25, 'quantity': 100}, {'name': 'Keyboard', 'price': 75, 'quantity': 50}]
```

سؤال ۵: مرتب‌سازی پیچیده لیست تودرتو

با توجه به تاپل USERS، تابعی بنویسید که این لیست را بر اساس جنسیت (sex) به صورت صعودی اول "Female" و بعد "Male" مرتب کند و در صورتی که جنسیت‌ها یکسان بود، کاربران را بر اساس سن (age) به صورت نزولی مرتب کند.

```
USERS = (  
    (1, "Nima Rabbani", "Male", 32),  
    (2, "FatemeH Rajabi", "Female", 28),  
    (3, "Hanieh Bahrami", "Female", 35),  
    (4, "Rashed Ragheb", "Male", 42)  
)  
  
# تابعی برای مرتب‌سازی بنویسید  
def sort_users_complex(users):  
    # کد شما اینجا  
    pass  
  
sorted_list = sort_users_complex(USERS)  
for user in sorted_list:  
    print(user)  
  
# خروجی مورد انتظار:  
# (3, 'Hanieh Bahrami', 'Female', 35)  
# (2, 'FatemeH Rajabi', 'Female', 28)  
# (4, 'Rashed Ragheb', 'Male', 42)  
# (1, 'Nima Rabbani', 'Male', 32)
```

سؤال ۶: پیش‌بینی خروجی (Shallow Copy)

با توجه به مفاهیم Shallow Copy که در فایل توضیح داده شد، خروجی نهایی دو دستور print در انتهای کد زیر چه خواهد بود؟ دلیل خود را در یک خط توضیح دهید.

```
import copy

original_list = [1, 2, [3, 4]]
shallow_copy_list = original_list.copy()

# تغییر در یک آیتم تودرتو
shallow_copy_list[2].append(5)

print("Original List:", original_list)
print("Shallow Copy List:", shallow_copy_list)
```

سؤال ۷: عملیات روی مجموعه‌ها (Set)

دو لیست از کاربران VIP و کاربران عادی در زیر وجود دارد. با استفاده از set، نام کاربرانی را پیدا کنید که فقط در یکی از دو لیست حضور دارند (اشتراکی نیستند).

```
vip_users = {"ali", "reza", "sara", "hadi"}
normal_users = {"nima", "sara", "hadi", "mahdi"}

# کد خود را برای یافتن کاربران غیرمشترک بنویسید
# ...

# خروجی مورد انتظار (ترتیب مهم نیست)
# {'ali', 'reza', 'nima', 'mahdi'}
```

سؤال ۸: دیکشنری شرطی

یک دیکشنری از نام محصولات و قیمت آنها وجود دارد. با استفاده از **Dictionary Comprehension**، یک دیکشن جدید بسازید که فقط شامل محصولاتی باشد که قیمت آنها بیشتر از ۵۰ دلار است و قیمت آنها در دیکشنری جدید با ۱۰٪ تخفیف محاسبه شود.

```
prices = {"Apple": 60, "Banana": 20, "Orange": 45, "Grapes": 80}

# discounted_prices = ...

print(discounted_prices)

# خروجی مورد انتظار:
# {'Apple': 54.0, 'Grapes': 72.0}
```