```
# Email Address Validator Using Regular Expressions (Regex)
# This project validates email addresses to check if they follow the correct format
# Purpose: Use regex pattern matching to verify valid email structure

# Step 1: Import the regex module
import re  # Import 're' (regular expression) library for pattern matching

email_condition = "^[a-z]+[\\._]?[a-z 0-9]+[@]\\w+[.]\\w{2,3}$"  # Define regex pattern: lowercase letters, optional dot/underscore, digits, @, d
user_email = input("Enter your Email: ")  # Get email input from user

if re.search(email_condition, user_email):  # Check if email matches the regex pattern
    print("Right Email")  # If match found, email is valid
else:  # If no match, email format is incorrect
    print("Wrong Email")
```

```
Enter your Email: mariam7@outlook.com
Right Email
```

```
# Test Case 2: Testing with invalid email (missing domain extension)
email_condition = "^[a-z]+[\\._]?[a-z 0-9]+[@]\\w+[.]\\w{2,3}$"
user_email = input("Enter your Email: ")

if re.search(email_condition, user_email):
    print("Right Email")
else:
    print("Wrong Email")
```

```
Enter your Email: mrnomi@hotmail
Wrong Email
```

```
# Test Case 3: Testing with invalid email (incomplete domain)
email_condition = "^[a-z]+[\\._]?[a-z 0-9]+[@]\\w+[.]\\w{2,3}$"
user_email = input("Enter your Email: ")

if re.search(email_condition, user_email):
    print("Right Email")
else:
    print("Wrong Email")
```

```
Enter your Email: abijaan@com
Wrong Email
```

```python
# Test Case 4: Testing with valid email (should pass validation)
email_condition = "^[a-z]+[\\._]?[a-z 0-9]+[@]\\w+[.]\\w{2,3}$"
user_email = input("Enter your Email: ")

if re.search(email_condition, user_email):
    print("Right Email")
else:
    print("Wrong Email")
```

```
Enter your Email: hira12@hotmail.com
Right Email
```