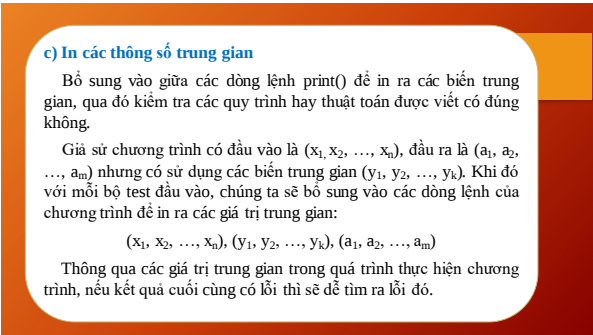
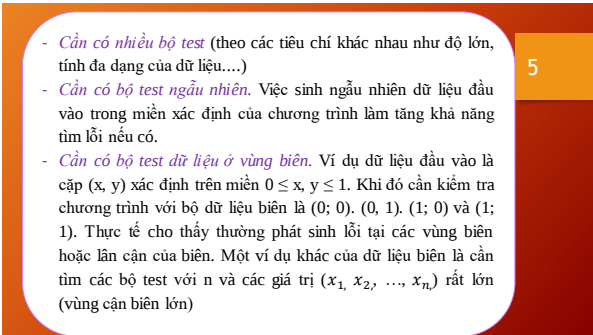
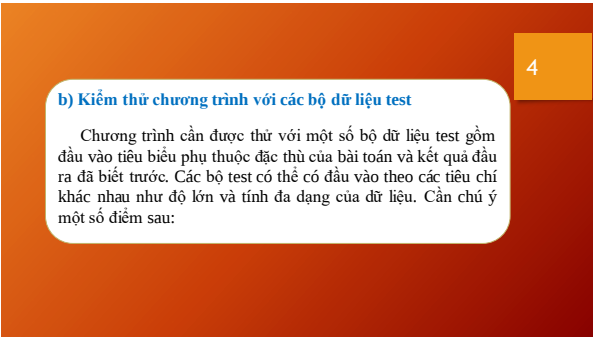
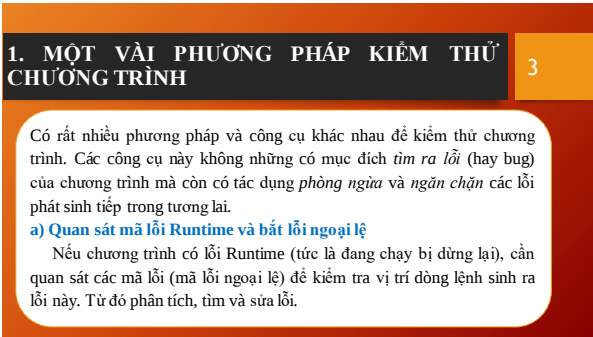
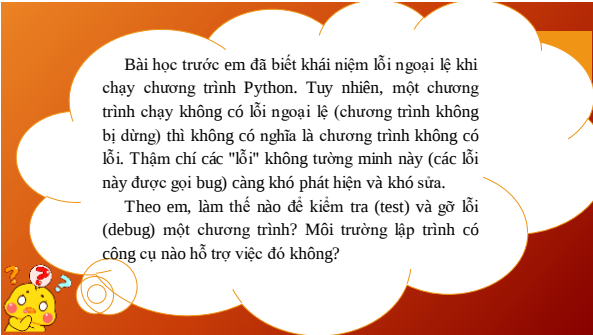




d) Sử dụng công cụ break point (điểm dừng)

Công cụ break point cho phép tạo ra các “điểm dừng” bên trong chương trình. Khi chạy, chương trình sẽ tạm dừng lại tại các “điểm dừng” cho phép người kiểm thử có thể quan sát các thông tin khác bên trong chương trình, qua đó kiểm tra tính đúng đắn của chương trình.

Trên thực tế sử dụng phương pháp điểm dừng thường kết hợp với phương pháp in các giá trị trung gian sẽ là hiệu quả hơn để kiểm thử chương trình.

Một số ghi nhớ:

- Sử dụng công cụ in các biến trung gian.
- Sử dụng công cụ sinh các bộ dữ liệu test.
- Sử dụng công cụ điểm dừng trong phần mềm soạn thảo lập trình.
- Quan sát các mã lỗi của chương trình nếu phát sinh.

2. VÍ DỤ MINH HỌA

Xét ví dụ sau: Nhập từ bàn phím hai số tự nhiên m, n, tính UCLN của hai số này.

Gọi gcd(m, n) là UCLN của hai số tự nhiên m, n. Thuật toán của bài toán này dựa trên bài toán sau:

- (1) gcd(m, m) = m.
- (2) Nếu n > m thì gcd(m, n) = gcd(m, n - m)
- (3) Nếu n < m thì gcd(m, n) = gcd(m - n, n).

Phần cơ bản nhất của chương trình sẽ là một vòng lặp while, vòng lặp sẽ kết thúc khi m = n. Chương trình như sau:

```
HoangThithanhTam_TL.py x
1 # Tính UCLN của m, n
2 m = int(input("Nhập số tự nhiên m: "))
3 n = int(input("Nhập số tự nhiên n: "))
4 while m != n:
5     k = k + 1
6     if m < n:
7         n = n - m
8     else:
9         m = m - n
10 print("Đáp số", m)
11
```

Chúng ta sẽ tiến hành kiểm thử chương trình này. Cần tập trung kiểm tra kĩ khối lệnh của lệnh lặp while

Cách 1: In ra các giá trị trung gian để kiểm soát chương trình. Bổ sung biến k và hai lệnh print() vào chương trình như mô tả như sau:

```
HoangThithanhTam_TL.py x
1 # Tính UCLN của m, n
2 m = int(input("Nhập số tự nhiên m: "))
3 n = int(input("Nhập số tự nhiên n: "))
4 k = 0
5 while m != n:
6     k = k + 1
7     print("Vòng lặp",k,":",m,n)
8     if m < n:
9         n = n - m
10    else:
11        m = m - n
12    print("Kết thúc vòng lặp ",m,n)
13    print("Đáp số", m)
14
```

```
>>> %Run HoangThithanhTam_TL.py
Nhập số tự nhiên m: 20
Nhập số tự nhiên n: 16
Vòng lặp 1 : 20 16
Vòng lặp 2 : 4 16
Vòng lặp 3 : 4 12
Vòng lặp 4 : 4 8
Kết thúc vòng lặp 4 4
Đáp số 4
```

Cách 2: Sử dụng công cụ tạo điểm dừng của phần mềm soạn thảo lập trình.

Thiết lập điểm dừng tại dòng 4 của chương trình như sau. Đây là vị trí bắt đầu chuẩn bị vào vòng lặp.

13

```
HoangThithanhTam_TL.py *
1 # Tính UCLN của m, n
2 m = int(input("Nhập số tự nhiên m = "))
3 n = int(input("Nhập số tự nhiên n = "))
4 while m != n:
5     k = k + 1
6     if m < n:
7         n = n - m
8     else:
9         m = m - n
10 print("Đáp số", m)
```

```
HoangThithanhTam_TL.py *
1 # Tính UCLN của m, n
2 m = int(input("Nhập số tự nhiên m = "))
3 n = int(input("Nhập số tự nhiên n = "))
4 k = 0
5 while m != n:
6     k = k + 1
7     if m < n:
8         n = n - m
9     else:
10        m = m - n
11 print("Đáp số", m)
```

14

Khi chạy chương trình sẽ dừng lại trước mỗi vòng lặp, chúng ta sẽ ghi lại các giá trị m, n vào một bảng như bảng sau. Khi kết thúc hết vòng lặp thì kết quả chương trình chính là giá trị m.

15

Vòng lặp	m	n	Kết quả
1	20	16	
2	4	16	
3	4	12	
4	4	8	
Kết thúc vòng lặp	4	4	4

Cả hai cách để kiểm soát lỗi là in các giá trị trung gian và thiết lập điểm dừng đều hiệu quả

16

LUYỆN TẬP

17

- 1. Chương trình của em khi chạy phát sinh lỗi ngoại lệ ZeroDivisionError. Đó là lỗi gì và em sẽ xử lý lỗi này như thế nào?
- 2. Chương trình sau có lỗi không? Nếu có thì tìm và sửa lỗi.

```
m = input (" Nhập số tự nhiên m: ")
n = input (" Nhập số tự nhiên n: ")
print (" Tổng hai số đã nhập là:" ,m+n)
```

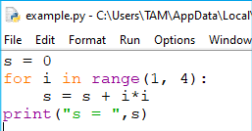
- 3. Chương trình sau có chức năng sắp xếp một dãy số cho trước. Hãy kiểm tra xem chương trình có lỗi không? Nếu có thì tìm và sửa lỗi.

```
A = [10, 1, 5, 2, 8, 0, 4]
for i in range ( len(A)-1):
    j = i
    while j > 1 and A[ j ] < A[ j - 1]:
        A[ j ], A[ j - 1] = A[ j - 1], A[ j ]
        j = j - 1
print(A)
```
- 4. Để kiểm thử một chương trình, nếu chỉ bằng việc kiểm tra thông qua các bộ dữ liệu test thì có bảo đảm tìm ra hết lỗi của chương trình hay không? Vì sao?

BÀI TẬP

19

- Em hãy soạn thảo và thực hiện từng bước chương trình ở hình sau



```
example.py - C:\Users\TAM\AppData\Local\
File Edit Format Run Options Window
s = 0
for i in range(1, 4):
    s = s + i*i
print("s = ", s)
```

BÀI TẬP

20

Câu 1: Em hãy nêu một vài lỗi thuộc nhóm lỗi cú pháp và một vài lỗi thuộc nhóm lỗi ngữ nghĩa

Câu 2: Tại sao phải tạo nhiều bộ dữ liệu vào khác nhau để kiểm thử chương trình?

Câu 3: Có bao nhiêu nhóm dữ liệu khác nhau cần tạo ra để kiểm thử chương trình?

Câu 4: Có thể xem giá trị các biến sau khi thực hiện một câu lệnh ở đâu?