

Sở GD&ĐT Thành phố Đà Nẵng

Trường THPT Đỗ Đăng Tuyển

Giáo viên soạn: Lâm Thị Hương

Ngày soạn: 30/03/2026

Lớp dạy: 10/1, 10/3, 10/7

Thời gian thực hiện: Tuần 30

## **CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH**

### **BÀI 30: KIỂM THỬ VÀ GỠ LỖI CHƯƠNG TRÌNH**

*Môn học: Tin học lớp 10. Thời gian thực hiện: 2 tiết*

#### **I. MỤC TIÊU**

##### **1. Về kiến thức**

- Biết được một vài phương pháp đơn giản kiểm thử chương trình.
- Biết được một vài cách gỡ lỗi đơn giản một chương trình.

##### **2. Về năng lực**

###### 2.1. Năng lực chung

- Năng lực tự chủ, tự học.
- Năng lực giao tiếp và hợp tác.
- Năng lực giải quyết vấn đề.

###### 2.2. Năng lực tin học:

- NLc: Giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông;
- Nle: Hợp tác trong môi trường số.

###### 2.3 Phát triển các năng lực số cốt lõi

###### 3.4.NC1a: HS tự thao tác để kiểm thử và gỡ lỗi được chương trình.

**3. Về phẩm chất:** Hình thành ý thức trách nhiệm, tính cẩn thận khi làm việc nhóm, phẩm chất làm việc chăm chỉ, chuyên cần để hoàn thành một nhiệm vụ

#### **II. THIẾT BỊ DẠY HỌC VÀ HỌC LIỆU SỐ**

##### **1. Đối với GV:**

- Chuẩn bị các bộ dữ liệu Test để kiểm thử chương trình.
- Phòng thực hành Tin học đã cài sẵn phần mềm Python.

##### **2. Đối với HS:**

- Biết được khái niệm về kiểm thử (test) và gỡ lỗi chương trình (debug).
- Biết một số phương pháp kiểm thử hay dùng trên thực tế.

- SGK Tin học 10, vở ghi chép nội dung bài đã soạn sẵn.

### III. TIẾN TRÌNH DẠY HỌC

#### A. HOẠT ĐỘNG MỞ ĐẦU (? phút)

##### a. Mục tiêu:

- Gợi ý cho học sinh hướng đến khái niệm kiểm thử (test) và gỡ lỗi một chương trình (debug).

##### b. Nội dung:

- Phân biệt được kiểm thử (test) có nghĩa là kiểm tra và chạy thử.
- Biết được lỗi tiếng Anh là bug (nghĩa là đen, là con bọ).
- Debug: de là lấy ra, bug là con bọ (gỡ lỗi của chương trình).

##### c. Sản phẩm:

- HS trả lời được:

1. Kiểm thử (test) có nghĩa là kiểm tra và chạy thử.
2. Debug là tìm bug và tháo bỏ bug ra khỏi chương trình

##### d. Tổ chức hoạt động

###### Bước 1. Giao nhiệm vụ học tập

- Cho HS quan sát một chương trình viết sẵn trên phần mềm python.

- Hỏi:

? Làm thế nào để kiểm tra (test) và gỡ lỗi (debug) một chương trình? Trong môi trường lập trình có công cụ nào hỗ trợ việc đó hay không?

###### Bước 2. Thực hiện nhiệm vụ: HS quan sát, nghiên cứu để trả lời câu hỏi của GV.

###### Bước 3. Báo cáo, thảo luận:

- HS thảo luận và trả lời được câu hỏi:

Để kiểm tra (test) và gỡ lỗi (debug) một chương trình có nhiều phương pháp, tiêu biểu như:

- Quan sát mã lỗi Runtime và bắt ngoại lệ
- Sử dụng nhiều bộ test dữ liệu
- In các thông số trung gian
- Tạo các điểm dừng để quan sát thông tin khác bên trong chương trình

Các công cụ hỗ trợ: công cụ in biến trung gian, sinh các bộ dữ liệu test, break point

###### Bước 4. Kết luận, nhận định:

- GV chạy thử chương trình để tìm lỗi và sử dụng một phương pháp để gỡ lỗi. Kết luận cuối cùng về câu hỏi đã đặt ra.

- Nhận định: Có nhiều phương pháp gỡ lỗi, tầm quan trọng của gỡ lỗi chương trình

## B. HOẠT ĐỘNG HÌNH THÀNH KIẾN THỨC

**Hoạt động 1:** Tìm hiểu một số phương pháp kiểm thử chương trình

a) **Mục Tiêu:** Học sinh biết được một số phương pháp kiểm thử hay dùng trên thực tế.

b) **Nội dung:** HS quan sát SGK để tìm hiểu nội dung kiến thức theo yêu cầu của GV

c) **Sản phẩm:** Hs hoàn thành tìm hiểu kiến thức

d) **Tổ chức thực hiện:**

HOẠT ĐỘNG GV - HS	DỰ KIẾN SẢN PHẨM
<p><b>Bước 1: Chuyển giao nhiệm vụ:</b></p> <p><b>GV:</b> Nêu đặt câu hỏi</p> <p>Đọc và thảo luận nhóm các phương pháp, công cụ sau để biết chức năng, tác dụng của từng công cụ trong công việc kiểm thử chương trình.</p> <p><b>HS:</b> Thảo luận, trả lời</p> <p><b>Bước 2: Thực hiện nhiệm vụ:</b></p> <p>+ HS: Suy nghĩ, tham khảo sgk trả lời câu hỏi</p> <p>+ GV: quan sát và trợ giúp các cặp.</p> <p><b>Bước 3: Báo cáo, thảo luận:</b></p> <p>+ HS: Lắng nghe, ghi chú, một HS phát biểu lại các tính chất.</p> <p>+ Các nhóm nhận xét, bổ sung cho nhau.</p> <p><b>Bước 4: Kết luận, nhận định:</b></p> <p>- GV chạy thử một chương trình còn lỗi và gỡ lỗi cho HS quan sát</p> <p>- Chính xác hóa và gọi 1 học sinh nhắc lại kiến thức</p>	<p><b>Sản phẩm 1:</b> Trả lời được câu hỏi của GV</p> <p>-Nếu chương trình có lỗi Runtime (tức là đang chạy bị dừng lại), cần quan sát các mã lỗi (mã lỗi ngoại lệ) để kiểm tra vị trí dòng lệnh sinh ra lỗi</p> <p><b>Sản phẩm 2:</b> Trả lời được câu hỏi của GV</p> <p>- Chương trình cần được thử với một số bộ dữ liệu test gồm đầu vào tiêu biểu phụ thuộc đặc thù của bài toán và kết quả đầu ra đã biết trước. Các bộ test có thể có đầu vào theo các tiêu chí khác nhau như độ lớn và tính đa dạng của dữ liệu.</p> <p><b>Sản phẩm 3:</b> Trả lời được câu hỏi của GV</p> <p>- Bổ sung vào giữa các dòng lệnh print() để in ra các biến trung gian, qua đó kiểm tra các quy trình hay thuật toán được viết có đúng không.</p> <p><b>Sản phẩm 4:</b> Trả lời được câu hỏi của GV</p> <p>- Công cụ break point cho phép tạo ra các “điểm dừng” bên trong chương trình. Khi chạy, chương</p>

HOẠT ĐỘNG GV - HS	DỰ KIẾN SẢN PHẨM
	<p>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.</p> <p><b>Sản phẩm 5:</b> HS theo dõi GV làm mẫu và ghi lại nội dung bài.</p>

## Hoạt động 2: Ví dụ minh họa

**a) Mục tiêu:** Giúp học sinh biết và có thể thực hiện được một số phương pháp kiểm thử đơn giản.

**b) Nội dung:** Gv giới thiệu bài toán minh họa là ƯCLN của 2 số tự nhiên  $m, n$  được nhập từ bàn phím.

**c) Sản phẩm:**

- In được các giá trị trung gian để kiểm soát chương trình.

- Sử dụng các công cụ tạo điểm dừng của phần mềm soạn thảo lập trình.

**d) Tổ chức thực hiện:**

HOẠT ĐỘNG CỦA GV - HS	DỰ KIẾN SẢN PHẨM
<p><b>Bước 1: Chuyển giao nhiệm vụ:</b></p> <p><b>GV:</b>  <b>HS:</b> Thảo luận, trả lời  <b>HS:</b> Lấy các ví dụ trong thực tế.</p> <p><b>Bước 2: Thực hiện nhiệm vụ:</b></p> <p>+ HS: Suy nghĩ, tham khảo sgk trả lời câu hỏi  + GV: quan sát và trợ giúp các cặp.</p> <p><b>Bước 3: Báo cáo, thảo luận:</b></p> <p>+ HS: Lắng nghe, ghi chú, một HS phát biểu lại các tính chất.  + Các nhóm nhận xét, bổ sung cho nhau.</p>	<p><b>Sản phẩm 1:</b></p> <p>Xét ví dụ sau: Nhập từ bàn phím hai số tự nhiên <math>m, n</math>, tính ƯCLN của hai số này.</p> <p>Gọi <math>\text{gcd}(m, n)</math> là ƯCLN của hai số tự nhiên <math>m, n</math>. Thuật toán của bài toán này dựa trên bài toán sau:</p> <p>(1) <math>\text{gcd}(m, m) = m</math>.  (2) Nếu <math>n &gt; m</math> thì <math>\text{gcd}(m, n) = \text{gcd}(m, n - m)</math>  (3) Nếu <math>n &lt; m</math> thì <math>\text{gcd}(m, n) = \text{gcd}(m - n, n)</math>.</p> <p>Chương trình như sau:</p> <pre> HoangThithanhTam_TL.py x 1 # Tính ƯCLN 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 &lt; n: 7         n = n - m 8     else: 9         m = m - n 10 print("Đáp số", m) 11 </pre>

## HOẠT ĐỘNG CỦA GV - HS

### Bước 4: Kết luận, nhận định:

- Phần cơ bản nhất của chương trình sẽ là một vòng lặp while, vòng lặp sẽ kết thúc khi  $m = n$ .

- Bổ sung thêm biến k và hai lệnh print() để in các giá trị trung gian k,m,n

- Quan sát sự thay đổi của các biến k, m, n trong quá trình thực hiện chương trình để phát hiện lỗi (nếu có), đồng thời hiểu được lỗi này và tìm cách sửa lỗi.

## DỰ KIẾN SẢN PHẨM

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
```

Kết quả thực hiện chương trình trên như sau:

```
>>> %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
```

### Sản phẩm 2:

**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.

```
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
```

Sửa lại

HOẠT ĐỘNG CỦA GV - HS	DỰ KIẾN SẢN PHẨM																								
<p>- Thiết lập điểm dừng tại dòng 4 của chương trình, đây là 1 vị trí bắt đầu một vòng lặp mới của lệnh while</p> <p>- Cả hai cách để kiểm soát lỗi là in các giá trị trung gian và thiết lập điểm dừng đều hiệu quả.</p>	<div data-bbox="869 212 1436 548" style="border: 1px solid gray; padding: 5px;"> <pre> HoangThithanhTam_TL.py x 1 # Tính ƯCLN 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 &lt; n: 8         n = n - m 9     else: 10        m = m - n 11 print("Đáp số", m) </pre> </div> <p>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.</p> <table border="1" data-bbox="821 761 1428 1232"> <thead> <tr> <th>Vòng lặp</th> <th>m</th> <th>n</th> <th>Kết quả</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>20</td> <td>16</td> <td></td> </tr> <tr> <td>2</td> <td>4</td> <td>16</td> <td></td> </tr> <tr> <td>3</td> <td>4</td> <td>12</td> <td></td> </tr> <tr> <td>4</td> <td>4</td> <td>8</td> <td></td> </tr> <tr> <td>Kết thúc vòng lặp</td> <td>4</td> <td>4</td> <td>4</td> </tr> </tbody> </table> <p><b>Sản phẩm 3:</b> Ghi hướng dẫn và theo dõi bài làm mẫu của GV.</p>	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
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																						



### C. HOẠT ĐỘNG LUYỆN TẬP (Thời gian ?)

**a . Mục tiêu:** Nhận ra lỗi trong chương trình và cách xử lý lỗi.

**b. Nội dung**

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)

```

**c. Sản phẩm**

- Bài làm của học sinh, kỹ năng giải quyết nhiệm vụ học tập.

**d. Tổ chức hoạt động**

**Bước 1. Giao nhiệm vụ học tập:**

- Thực hiện các bài tập 1, 2 SGK trang 148

**Bước 2. Thực hiện nhiệm vụ:**

- HS thực hiện theo yêu cầu

**Bước 3. Báo cáo, thảo luận:**

- Kết thúc thảo luận, GV kiểm tra sản phẩm của HS.

**Bước 4. Kết luận, nhận định:**

- GV kiểm tra, đánh giá sản phẩm và cho điểm theo mức độ hoàn thành nhiệm vụ của HS (có thể đánh giá theo từng sản phẩm hoặc toàn bộ sản phẩm của bài).

- Hướng dẫn HS sửa lỗi (nếu có) theo các nội dung sau:

**Bài 1:** Đây là lỗi chia cho 0. Nếu gặp lỗi này cần tìm ra vị trí gây lỗi để biết phép tính nào tạo ra phép chia cho 0. Sau đó sẽ tiến hành các biện pháp xử lý lỗi ngay trong chương trình, ví dụ chỉ cho phép thực hiện phép chia khi mẫu số khác 0.

**Bài 2:** Chương trình sau có lỗi do khi nhập dữ liệu từ bàn phím chưa thực hiện chuyển đổi sang số nguyên. Cần sửa 2 lệnh đầu như sau:

```
m = int(input ( " Nhập số tự nhiên m: "))
```

```
n = int(input ( " Nhập số tự nhiên n: "))
```

**GV lưu ý:** Có thể Hướng dẫn HS thực hiện trên lớp hoặc ở nhà:

+ GV có thể đưa thêm các ví dụ tương tự để học sinh thực hiện việc kiểm lỗi và xử lý lỗi trong chương trình đơn giản.

**D. HOẠT ĐỘNG VẬN DỤNG (Thời gian ?)**

**a) Mục tiêu:** Vận dụng các kiến thức vừa học quyết các vấn đề học tập và thực tiễn.

**b) Nội dung:**

Bài 1: 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)
```

Bài 2: 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?

**c) Sản phẩm:** HS vận dụng các kiến thức trả lời 2 câu hỏi trong SGK trang 148.

**d) Tổ chức thực hiện:**

**Bước 1. Giao nhiệm vụ học tập:**

- Thực hiện các bài tập 1, 2 SGK trang 148

**Bước 2.** *Thực hiện nhiệm vụ:*

- HS thực hiện theo yêu cầu

**Bước 3.** *Báo cáo, thảo luận:*

- Kết thúc thực hành, GV kiểm tra sản phẩm của HS.

**Bước 4.** *Kết luận, nhận định:*

- GV kiểm tra, đánh giá sản phẩm và cho điểm theo mức độ hoàn thành nhiệm vụ của HS (có thể đánh giá theo từng sản phẩm hoặc toàn bộ sản phẩm của bài).

- Hướng dẫn HS sửa lỗi (nếu có) theo các nội dung sau:

**Bài 1:** Chương trình có lỗi tại dòng lệnh thứ 4 lệnh: (while  $j > 1$  and  $A[j] < A[j - 1]$ ;) )

Sửa lại dòng này như sau: while  $j > 0$  and  $A[j] < A[j - 1]$ :

**Bài 2:** Không đảm bảo hết lỗi. Việc kiểm tra chương trình bằng các bộ dữ liệu test càng nhiều sẽ càng làm tăng độ tin cậy của chương trình nhưng vẫn chưa đảm bảo tìm ra hết lỗi của chương trình, hay nói cách khác chưa chứng minh được rằng chương trình đúng.