

# Lỗi

---

Nguyễn Tô Sơn

<http://hochtincungthukhoa.com>

Khi lập trình để giải quyết một bài toán trên máy tính, điều không tránh khỏi của chúng ta là gặp lỗi. Ta chia lỗi thành ba loại: Lỗi cú pháp, lỗi xảy ra trong quá trình thực thi và lỗi thuật toán.

Đối với lỗi cú pháp và lỗi xảy ra trong quá trình thực thi, ta có thể sửa ngay lập tức bởi hệ thống phát hiện lỗi phong phú của ngôn ngữ lập trình. Còn đối với lỗi thuật toán, công việc không hề đơn giản như vậy.

Chúng ta cần phân biệt: lỗi xảy ra trong quá trình thực thi và lỗi thuật toán.

Lỗi xảy ra trong quá trình thực thi là những lỗi như: chia cho 0, hoặc căn bậc hai của một số âm, tính tan của  $\pi/2$ ... Lỗi này không phải lỗi cú pháp nên không được hệ thống báo lỗi phát hiện trước khi thực thi chương trình. Vì vậy, để tránh những lỗi nhằm đỡ mất thời gian phải tùy thuộc vào sự kiểm soát của người lập trình.

Lỗi thuật toán là lỗi làm cho chương trình thực hiện sai lệch so với thuật toán đề ra. Đây là lỗi nguy hiểm nhất khi lập trình. Một chương trình hoàn chỉnh, không có lỗi thuật toán là sản phẩm của nghệ thuật giải quyết những lỗi dạng này, nó phụ thuộc rất nhiều vào trình độ của người lập trình.

Ta có thể lấy ví dụ đơn giản về lỗi thuật toán: Viết chương trình sắp xếp dãy số theo thứ tự giảm dần, nhưng chương trình lại sắp xếp theo thứ tự tăng dần. Nguyên nhân được phát hiện ở đây là do “sự tiện tay” khi viết chương trình của người lập trình, những yếu tố chủ quan được xác định trong đó. Tuy nhiên, trong thực tế viết chương trình, công việc phức tạp hơn nhiều.

Rất may mắn, các ngôn ngữ lập trình đều có hệ thống để trợ giúp lập trình viên phát hiện ra những lỗi nguy hiểm này. Nó được đặt tên đúng như ý nghĩa của nó: Gỡ rối (Debug). Tuy nhiên, việc giảng dạy tại các nhà trường phổ thông do không đủ thời lượng, cũng như khả năng hạn chế của học sinh nên không đưa vào giảng dạy hệ thống gỡ rối này.

Bài viết này giới thiệu về hệ thống gỡ rối trong ngôn ngữ Turbo Pascal và cách sử dụng chúng một cách hiệu quả. Đối với các ngôn ngữ khác, việc thực hiện là tương tự.

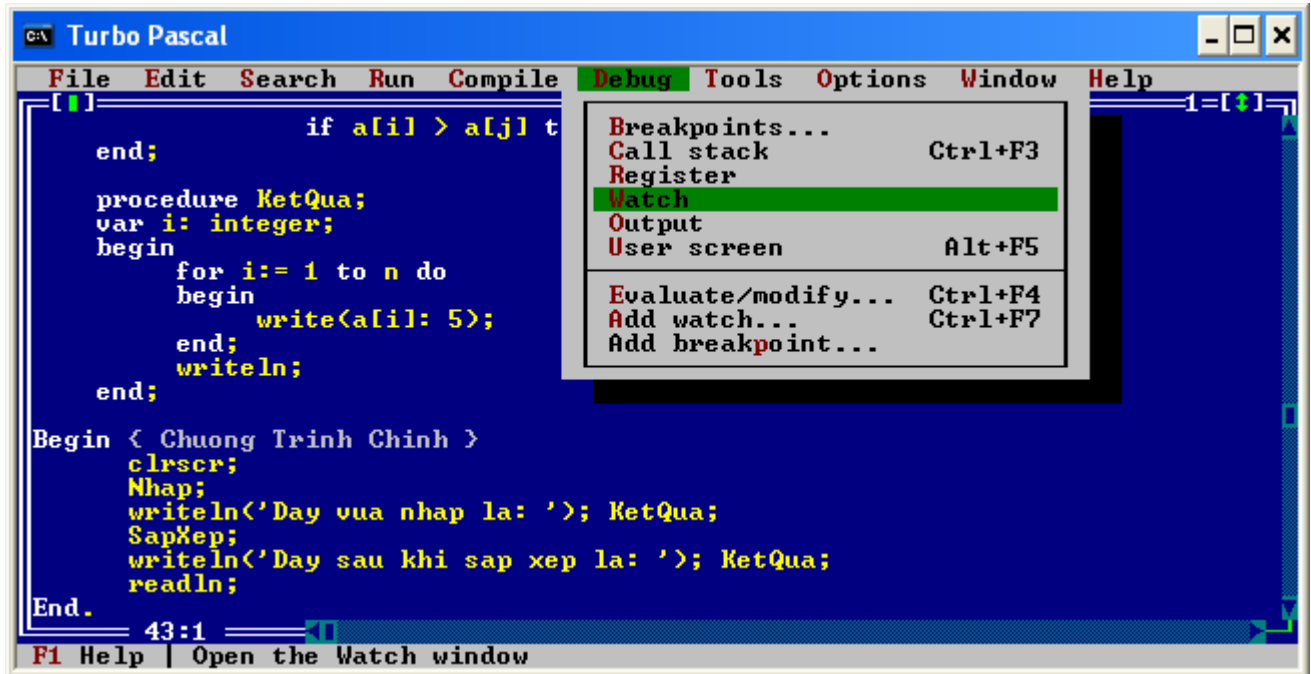
Mặc dù hiện nay Turbo Pascal là miễn phí nhưng cũng khuyên các bạn sử dụng phần mềm cài đặt đầy đủ, hoặc đã được sửa lỗi để tránh những lỗi không cần thiết trong quá trình sử dụng.

**Lời khuyên: download bộ cài trên DOS để cài đặt.**

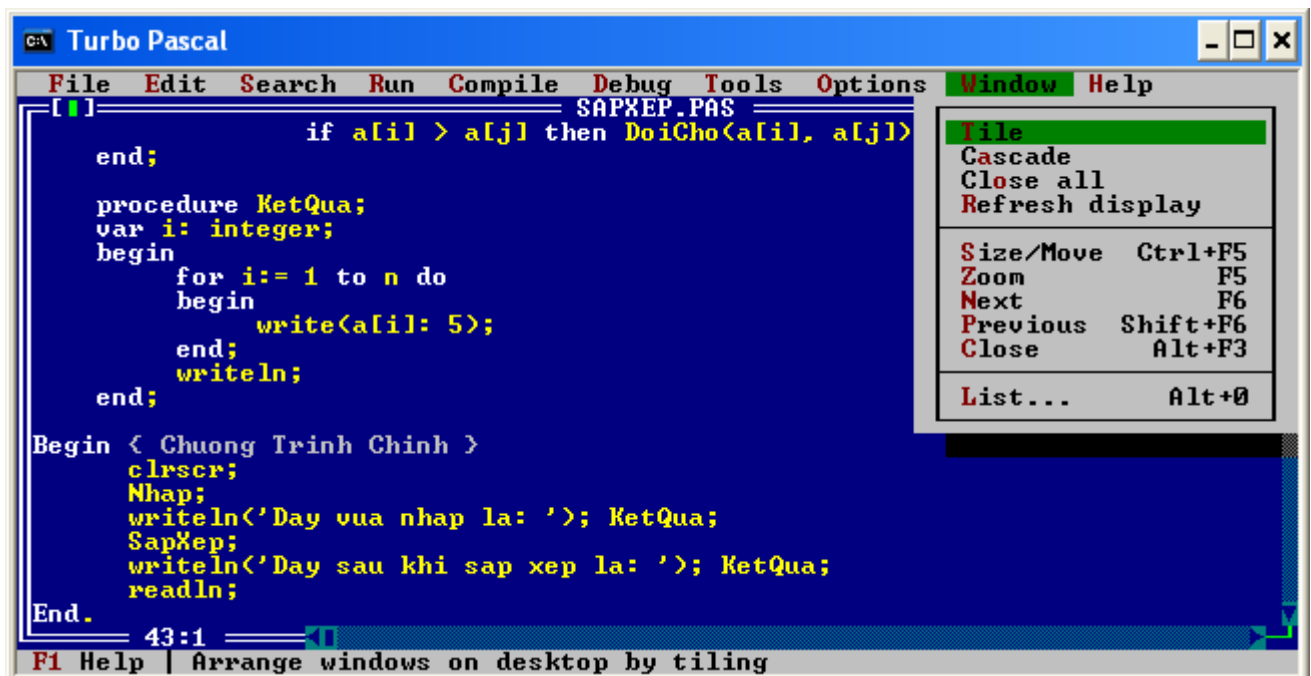
## Giới thiệu về hệ thống gỡ rối Debug

Để sử dụng hệ thống báo lỗi ta thực hiện:

- Trên thanh Menu: Vào Debug/Watch để bật cửa sổ hiển thị biến trong quá trình thực hiện.



- Để cửa sổ Watch không biến mất: Trên thanh Menu, vào Window/Title



- Để gỡ rối lỗi thuật toán ta kiểm soát giá trị của các biến từng bước để xem nó thực hiện đúng hay sai ở từng bước, từ đó, đưa ra những quyết định đúng đắn để sửa đổi. Để thêm

một biến vào cửa sổ Watch: Trên thanh Menu vào Debug/Add watch hoặc sử dụng tổ hợp phím nóng: Ctrl + F7. Sau đó, gõ tên biến cần quan sát giá trị vào.



- Bạn có thể xem tại cửa sổ Watch giá trị của bất cứ biến loại nào: integer, char, mảng một chiều, mảng hai chiều, record... Riêng đối với mảng hai chiều bạn còn có thể xem giá trị của một hàng nào đó trong mảng đó.

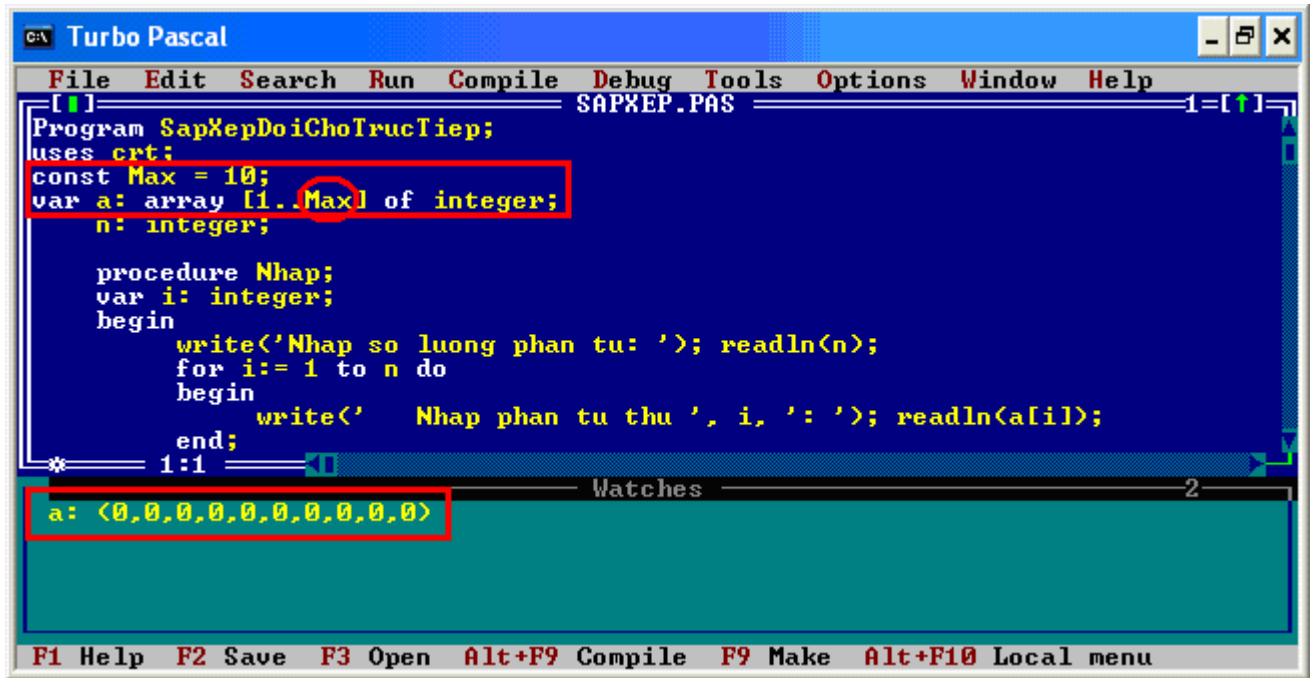
Ví dụ mảng hai chiều:

```
mang2: array [1..100][1..200] of integer;
```

Để xem giá trị hàng thứ 10 trong cửa sổ Watch ta có thể gõ trực tiếp khi Add watch: mang2[10]

- Một điều cũng vô cùng quan trọng khi gỡ rối là sử dụng cách sử dụng const số lượng của các mảng để cửa sổ Watch dễ nhìn hơn. Bạn nên khai báo số lượng phần tử khai báo trong mảng bằng các const, khi gỡ rối thì sửa lại là số lượng phần tử các lần thử nghiệm. Đến lúc thực thi thật thì mới thay bằng số lượng phần tử tối đa của mảng.

Hình vẽ sau đây thể hiện điều này, chú ý tới các vị trí đánh dấu màu đỏ.



- Sử dụng các kiểu biên dịch chương trình (Thứ tự sử dụng giảm dần khi gõ rồi)
  - F7: Dịch theo từng dòng
  - F8: Dịch theo từng bước được chỉ ra trong chương trình chính
  - F4: Dịch từ đầu đến vị trí con trỏ đang đứng
  - F9: Dịch toàn bộ chương trình
  - Ctrl + F9: Dịch toàn bộ chương trình và chạy

