



Bài thực hành 4

SỬ DỤNG CÂU LỆNH ĐIỀU KIỆN

Thời lượng: 2 tiết

1. Mục đích, yêu cầu

- Viết được câu lệnh điều kiện *if...then* trong chương trình;
- Rèn luyện kỹ năng ban đầu về đọc các chương trình đơn giản và hiểu được ý nghĩa của thuật toán sử dụng trong chương trình.

2. Những điểm cần lưu ý và gợi ý dạy học

a) Bài này là bài đầu tiên HS thực hành sử dụng lệnh *if... then*. Do vậy, các ví dụ cần đơn giản, dễ hiểu để HS dễ dàng nhận ra ý nghĩa, hoạt động của câu lệnh điều kiện, biểu thức điều kiện đơn giản (phép so sánh), câu lệnh trong cấu trúc rẽ nhánh.

HS đã được làm quen với thuật toán này ở bài tập 4, bài 5. Do vậy, nói chung HS sẽ không gặp khó khăn về thuật toán này khi tìm hiểu chương trình *Sap_xep* ở bài 1 của bài thực hành này. Hoàn toàn có thể sử dụng chương trình *Sap_xep* để đạt mục tiêu thực hành sử dụng câu lệnh *if... then...else*.

Tuy nhiên, dưới đây xin giới thiệu một phương án về cơ bản vẫn dựa trên các yêu cầu của bài 1 nhưng có chỉnh sửa đôi chút để GV tham khảo khi tiến hành dạy học.

b) Giữ nguyên yêu cầu đề bài của bài 1. Đối với câu a, yêu cầu HS mô tả các bước để giải quyết bài toán. Các bước cơ bản để giải bài toán này là:

Bước 1. Nhập hai số nguyên a, b từ bàn phím;

Bước 2. Nếu $a \leq b$ thì hiển thị ra màn hình giá trị biến a trước rồi đến giá trị biến b ;

Bước 3. Nếu $b < a$ thì hiển thị ra màn hình giá trị biến b trước rồi đến giá trị biến a ;

Bước 4. Kết thúc.

Trên cơ sở phân mô tả thuật toán, GV hướng dẫn để HS viết được chương trình tương ứng. Chương trình có thể như sau:

```
program Sap_xep;  
uses crt;
```

```

var A, B, T: integer;
begin
  clrscr;
  {Buoc 1: Nhap hai so nguyen a, b tu ban phim}
  write('Nhap so A:'); readln(A);
  write('Nhap so B:'); readln(B);
  {Buoc 2: Neu a < b thi hien thi ra man hinh gia tri bien a
  truoc roi den gia tri bien b}
  if A<=B then write(A, ' ',B);
  {Neu b < a thi hien thi ra man hinh gia tri bien b truoc roi
  den gia tri bien a}
  if B<A then write(B, ' ',A);
  readln
End.

```

c) Thuật toán thực hiện trong chương trình này có thể sẽ gần với cách nghĩ của HS hơn. Do vậy, hi vọng HS sẽ thấy gần gũi và dễ dàng hơn trong việc tìm hiểu chương trình này cũng như hiểu được hoạt động, cách sử dụng lệnh rẽ nhánh trong chương trình này. Hơn nữa, về mặt sự phạm nên giới thiệu câu lệnh điều kiện dạng thiếu trước vì câu lệnh này đơn giản hơn câu lệnh điều kiện dạng đủ. Sau khi giới thiệu câu lệnh dạng thiếu, việc giới thiệu câu lệnh dạng đủ sẽ thuận lợi hơn.

Yêu cầu HS chạy chương trình và thử với các bộ số liệu có trong câu c, bài 1 (SGK).

d) Có một số điểm chính GV cần lưu ý khi cho HS tiến hành bài 2 như sau:

- Với câu lệnh điều kiện dạng đủ *if...then...else*, lưu ý không đặt dấu chấm phẩy sau câu lệnh trước từ khoá *else*. Trong Pascal, dấu chấm phẩy dùng để phân cách các câu lệnh (không phải là kết thúc câu lệnh).

- Đoạn chương trình:

```

If Long>Trang then writeln('Ban Long cao hon');
If Long<Trang then writeln('Ban Trang cao hon')
                else writeln('Hai ban cao bang nhau');

```

Thoạt nhìn, có thể theo cách suy luận của nhiều HS thì sự kết hợp của hai câu lệnh điều kiện là đảm bảo đầy đủ các trường hợp, đặc biệt câu lệnh thứ nhất đã xét đến trường hợp *Long>Trang* nên câu lệnh thứ hai sẽ không còn xét đến trường hợp này nữa mà chỉ xét đến hai trường hợp còn lại là *Long<Trang* và *Long = Trang*. Cách suy luận như vậy không phải là không có lí khi xét

trong trường hợp con người xử lí tình huống này. Tuy nhiên, máy tính không xử lí như vậy.

Khi máy tính thực hiện đến câu lệnh thứ hai là

```
If Long<Trang then writeln('Ban Trang cao hon')
                    else writeln('Hai ban cao bang nhau');
```

thì máy tính lại xét tất cả các trường hợp có thể. Vì vậy, đã dẫn đến lỗi thực hiện chương trình trong trường hợp Long cao hơn Trang.

e) Dưới đây xin phân tích rõ hơn về đoạn chương trình này, GV xem xét, tham khảo để giảng dạy cho HS. GV không cần dành quá nhiều thời gian giải thích quá chi tiết, tỉ mỉ như trình bày dưới đây.

Khi thực hiện đến câu lệnh thứ hai, máy tính thử kiểm tra điều kiện $Long < Trang$ và có các trường hợp xảy ra như sau:

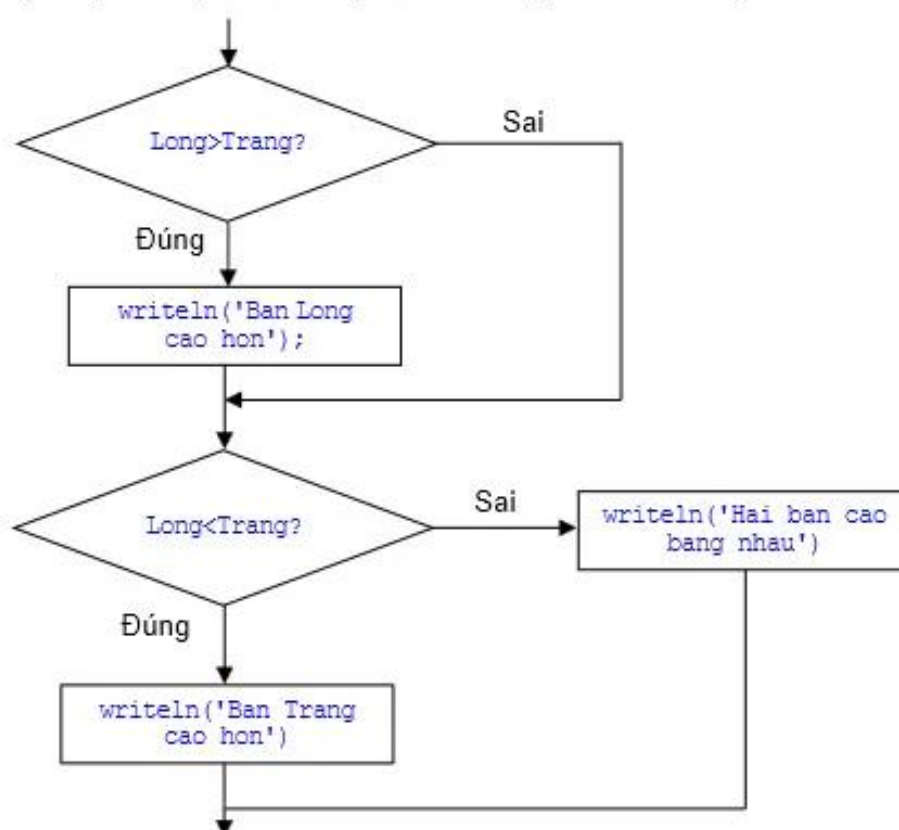
- Nếu $Long < Trang$ cho kết quả đúng, tức là Trang cao hơn Long thì máy hiển thị ra màn hình dòng chữ "*Ban Trang cao hon*". Trong trường hợp này thì câu lệnh trước là

```
If Long>Trang then writeln('Ban Long cao hon');
```

đã không hiển thị ra màn hình dòng chữ "*Ban Long cao hon*" bởi vì biểu thức điều kiện $Long > Trang$ trong câu lệnh điều kiện thiếu này cho kết quả sai. Kết quả là màn hình chỉ hiển thị một dòng chữ "*Ban Trang cao hon*".

- Nếu $Long < Trang$ cho kết quả sai, Pascal thực hiện câu lệnh 2 in ra màn hình dòng chữ "*Hai ban cao bang nhau*". Tuy nhiên, có hai trường hợp dẫn đến biểu thức $Long < Trang$ cho kết quả sai:
 - + Long và Trang cao bằng nhau, khi đó biểu thức điều kiện $Long > Trang$ cũng cho kết quả sai và câu lệnh điều kiện thứ nhất không hiển thị ra màn hình dòng chữ nào cả. Kết quả là màn hình chỉ có một dòng chữ "*Hai ban cao bang nhau*".
 - + Long cao hơn Trang, khi đó biểu thức điều kiện $Long > Trang$ ở câu lệnh điều kiện thứ nhất cho kết quả đúng. Vì vậy, câu lệnh điều kiện thứ nhất đã hiển thị ra màn hình dòng chữ "*Ban Long cao hon*". Như vậy, màn hình sẽ hiển thị ra hai dòng thông báo khác nhau là "*Ban Long cao hon*" và "*Hai ban cao bang nhau*". Đây chính là một lỗi ngữ nghĩa của chương trình cần được chỉnh sửa.

Sơ đồ thực hiện hai lệnh trên được mô tả trong hình dưới đây:



Về cách dạy phần này, nên cho HS gõ chương trình vào máy và chạy thử với các bộ dữ liệu kiểm tra. Các bộ số liệu cần phủ kín các trường hợp: Trang cao hơn Long, hai bạn cao bằng nhau và Long cao hơn Trang. Yêu cầu HS quan sát kết quả để phát hiện vấn đề và tìm hiểu chương trình, phát hiện lỗi.

f) Thực ra, việc thử chương trình với một số bộ dữ liệu mẫu là một trong các công đoạn của phát triển phần mềm, còn gọi là bước kiểm thử. Sau khi lập trình xong, phần mềm cần được thử nghiệm với một số bộ dữ liệu mẫu (hay còn gọi là bộ test) mà người ta dễ dàng biết được kết quả để kiểm chứng với kết quả mà chương trình đưa ra.

Việc thử chương trình với một bộ dữ liệu test chỉ chứng minh được chương trình sai mà không chứng minh được chương trình là đúng. Nghĩa là khi thử với bộ dữ liệu test nếu chương trình cho kết quả sai khác với kết quả đã được biết trước thì kết luận là chương trình sai, nhưng nếu thử với một số bộ dữ liệu test mà chương trình cho kết quả đúng với kết quả đã được biết trước thì chưa thể kết luận chương trình hoàn toàn đúng đắn được mà chỉ có thể nói

chưa phát hiện ra sai sót của chương trình mà thôi. Tuy nhiên, người ta có thể có cách khác để chứng minh tính đúng đắn của chương trình mà không chỉ sử dụng đến các bộ dữ liệu test.

Có hai cách để chỉnh sửa chương trình trên để đảm bảo chỉ đưa ra một thông báo đúng.

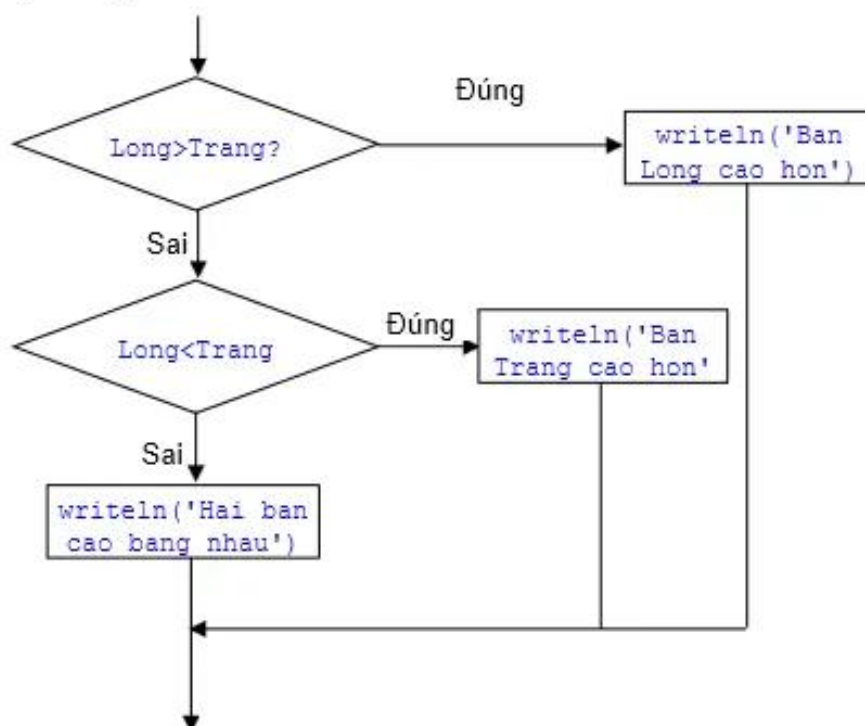
Cách đơn giản nhất là sử dụng ba câu lệnh điều kiện dạng thiếu như sau:

```
If Long>Trang then writeln('Ban Long cao hon');  
If Long=Trang then writeln('Hai ban cao bang nhau');  
If Long<Trang then writeln('Ban Trang cao hon');
```

Cách thứ hai là sử dụng câu lệnh điều kiện lồng nhau như trong SGK, mục đích của cách này là giới thiệu cho HS có thể sử dụng các câu lệnh điều kiện lồng nhau:

```
If Long>Trang then writeln('Ban Long cao hon') else  
If Long<Trang then writeln('Ban Trang cao hon')  
else writeln('Hai ban cao bang nhau');
```

Có thể sử dụng sơ đồ dưới đây để giải thích về hoạt động của hai câu lệnh điều kiện lồng nhau để HS hiểu được lí do câu lệnh này chỉ hiển thị ra màn hình một thông báo.



g) Với bài 3, HS cần biết điều kiện để ba số dương a, b, c là độ dài ba cạnh của một tam giác thì tổng hai cạnh phải lớn hơn cạnh còn lại, nghĩa là phải đồng thời thoả mãn ba điều kiện $a + b > c, b + c > a$ và $c + a > b$. GV hướng dẫn HS về cách biểu diễn ba điều kiện này trong Pascal:

$a+b>c$) and ($b+c>a$) and ($c+a>b$)

Điểm khó của bài này là HS biết chuyển biểu thức điều kiện toán học sang biểu diễn trong Pascal. Cần giải thích HS hiểu dùng phép quan hệ *and* là để đảm bảo cả ba điều kiện $a + b > c, b + c > a$ và $c + a > b$ đồng thời thoả mãn. Việc phải sử dụng dấu ngoặc đơn trong phép so sánh trên là để đảm bảo thứ tự ưu tiên thực hiện phép toán và để đảm bảo tham số của phép *and* (và *or*) chỉ có thể là giá trị đúng hoặc sai (không là số).

Cần cho HS đọc, thảo luận kĩ để hiểu chương trình này. Bài toán này là một trong các bài toán yêu cầu trong chuẩn kiến thức, kĩ năng. Sau khi học bài này, HS phải hiểu và phải tự viết chương trình giải bài toán tương tự (ví dụ kiểm tra tính chất (cân, đều, vuông) của tam giác dựa trên số đo của các cạnh).

Để HS luyện tập thêm về câu lệnh điều kiện, phép so sánh, có thể yêu cầu HS viết chương trình cho phép nhập điểm bài kiểm tra của một bạn nào đó, rồi thực hiện:

- Nếu điểm nhỏ hơn 5, in ra dòng chữ "*Ban can co gang hon*";
- Nếu điểm lớn hơn hoặc bằng 5 và nhỏ hơn 6.5, in ra dòng chữ "*Ban dat diem trung binh*";
- Nếu điểm lớn hơn hoặc bằng 6.5 và nhỏ hơn 8, in ra dòng chữ "*Ban dat diem kha*";
- Nếu điểm lớn hơn hoặc bằng 8, in ra dòng chữ "*Hoan ho, ban dat diem gioi*".

Một ví dụ khác là yêu cầu HS viết chương trình giải phương trình bậc nhất $ax + b = 0$, với $a \neq 0$ và a, b nhập từ bàn phím. Lưu ý, phương trình bậc nhất một ẩn số được giới thiệu trong chương trình môn Toán lớp 8 (học kì II). Do vậy, nếu HS chưa được học phương trình này ở môn Toán, GV nên dành một vài phút để HS làm quen với khái niệm này.