Dạy lệnh lặp với số lần biết trước trong ngôn ngữ lập trình Pascal

PGS TS Lê Khắc Thành, Khoa CNTT ĐHSP HN

1. Hình thành câu lệnh

Tình huống công việc dẫn đến lặp For.

Cho máy tính giá trị của biểu thức S = 2 + 3 + 4 + . . . + 21 + 22 + 23

Lần đầu tiên giới thiệu, hình thành vòng For cho học sinh, theo chúng tôi, chọn tình huống trên là thích hợp, nhất là khi dạy có máy tính cho học sinh quan sát ngay khi máy thực hiện chương trình. Đây không phải là người tính biểu thức trên mà là máy tính biểu thức trên. Giầu trí tò mò, thế nào các em học sinh cũng nhẩm kết quả để xem máy có tính đúng không? Việc nhẩm kết quả của biểu thức trên là rất dễ. Ta hướng dẫn cho các em biết tổng trên có 22 số hạng, mà hai số hạng cách đều đầu và cuối dãy có tổng là 25. Dãy trên có 11 cặp số có tổng là 25 vậy đáp số là 25 × 11 = 275 (nhân nhẩm 25 với 11, cộng 2 với 5 được 7 viết vào giữa 2 và 5). Khi máy cũng cho đáp số là 275 thì các em sẽ tin câu lệnh là đúng.

Nếu chúng ta cho biểu thức cồng kềnh, khó nhẩm thì các em sẽ sa vào tính toán không dùng máy để xem máy có làm đúng không, điều đó làm mất thời gian không cần thiết, vì một tiết trên lớp chỉ có 45 phút cho thày và trò làm việc.

Một cách tính khác, cộng lần lượt theo thứ tự từ trái qua phải. Giáo viên mô phỏng cho học sinh quá trình cộng dồn theo biến K lần lượt nhận các giá trị từ 2 đến 23, mỗi lần K nhận một giá trị thì cộng K với S rồi lại gửi tổng vào biến S như sau S := S + K ;

K	S + K	S
		0
2	0 + 2	2
3	2 + 3	5
4	5 + 4	9
5	9+5	14
6	14 + 6	20
...	...	...
22	...	252
23	252 + 23	275

Như vậy việc lấy S + K rồi gán cho S được thực hiện 22 lần tuần tự theo các giá trị mà biến K lần lượt duyệt qua từ 2 đến 23.

GV giới thiệu Passcal có lệnh cho phép làm điều giống như việc mô phỏng ở trên.

Cú pháp

FOR biến := biểu thức 1 TO biểu thức 2 DO câu lệnh ;

Hoạt động : Từ biểu thức 1 đến biểu thức 2 biến nhận bao nhiêu giá trị thì câu lệnh được thực hiện bấy nhiêu lần.

Nhìn vào bảng mô phỏng trên, chúng ta thấy: Việc thêm dần K vào S làm cho S lần lượt nhận các giá trị 0, 2, 5, 9, 14, 20... , trong kiểu Byte cho đến khi K là 22 thì S là 252 vẫn thuộc kiểu Byte. Nhưng khi K là 23 thì S là 275 vượt ra ngoài phạm vi kiểu Byte sang kiểu Word. Như vậy biến S đã có sự thay đổi về lượng dẫn đến thay đổi về chất.

Việc K lần lượt nhận các giá trị từ 2 đến 23 nên ta khai K có kiểu Byte. Việc S nhận giá trị cuối cùng là 275 vượt ra ngoài phạm vi kiểu byte vậy biến S phải khai kiểu Word.

Việc khai báo biến S chứa tổng có kiểu Word là một dịp để giáo viên nhắc nhở học sinh quan tâm đến phương diện dữ liệu khi khai báo biến để tránh tràn ô nhớ (việc tràn ô nhớ xảy ra ngay khi thực hiện phép tính trên thanh ghi chứ không phải là gán vào mới tràn). Từ đó dẫn đến chương trình như sau:

```
Program  ViDu ;   Uses crt;
Var      K : Byte ;  S :  Word ;
Begin
   ClrScr;   S := 0 ;
   For   K := 2 TO  23  DO    S := S + K ;
   Writeln ( ' Dap so la   ' , S ) ;
   Readln
End.
```

Chúng ta nhắc học sinh, biến S có tham gia vào biểu thức, nên việc khởi trị cho S là cần thiết. Để không làm ảnh hưởng đến tổng, thì ta khởi trị cho S nhận giá trị ban đầu bằng 0 thông qua lệnh gán S := 0 ;

2. Luyện tập lặp với số lần định trước

Sau khi hình thành cho học sinh vòng lặp For. Để luyện tập ta có thể cho học sinh làm bài với tình huống công việc như sau (để học sinh tâm phục khẩu phục, vì ví dụ trên dễ, nhẩm nhanh ra được đáp số ngay).

Tính S = $\sqrt{23+\sqrt{22+...+\sqrt{3+\sqrt{2}}}}$

Bài toán này có 22 dấu căn bậc hai lồng nhau. Việc thực hiện phải lần lượt tính từ trong ra ngoài. Đầu tiên tính $\sqrt{2}$ được 1.41, tiếp đến cộng 1.41 với 3 được 4.41 rồi lấy $\sqrt{4.41}$ kết quả này lại được cộng với 4 rồi lại lấy căn. Quá trình cứ tiếp tục như vậy. Chúng ta cho biến K lần lượt nhận các giá trị từ 2 đến 23, ứng với mỗi giá trị của k chúng ta lấy $\sqrt{S+K}$ gửi vào S.

Chúng ta dán chương trình trên sang cửa sổ thứ hai rồi sửa lại để có chương trình giải bài toán này (vừa sửa vừa giải thích lí do, chẳng hạn căn cho giá trị thực vậy nội dung lưu giữ ở biến S đã thay đổi nên S phải khai lại là Real cho phù hợp với dữ liệu mới, nội dung đã thay đổi thì hình thức phải thay đổi theo). Chương trình là như sau.

```
Program  ViDu ;   Uses crt;
Var   K : Byte ;  S :  Real ;
Begin
   ClrScr;   S := 0 ;
   For   K := 2 TO  23  DO    S := SQRT(S + K) ;
   Writeln ( ' Dap so la   ' ,  S : 0 : 4 ) ;
   Readln
End.
```

Lời bàn về phương pháp giảng dạy

- Về phương diện giao tiếp, khi tạo tình huống dạy vòng For, chúng ta cần một biến lần lượt duyệt qua các giá trị liên tục, đếm được. Việc thực hiện một công việc nào đó được lặp đi, lặp lại đúng bằng số phần tử của tập giá trị mà biến duyệt qua. Chúng ta cho tính tổng các số tự nhiên từ 2 đến 23 là còn nhằm đến các phương

diện dữ liệu, thuật toán, kết quả, quá trình và nhằm rèn một số thao tác tư duy cho học sinh và nhằm bồi dưỡng quan điểm triết học cho họ.

Mục đích chức năng của chúng ta là hình thành vòng For cho học sinh, mục đích phương tiện là tính tổng các số tự nhiên từ 2 đến 23. Phương châm của chúng ta là thực hiện mục đích chức năng thông qua mục đích phương tiện. Vì vậy mà ví vụ minh hoạ (mục đích phương tiện) chúng ta chọn càng đơn giản về mặt tính toán, càng dễ hiểu về mặt cấu trúc càng tốt. Việc tính tổng từ 2 đến 23 chỉ là phương tiện trực quan mô phỏng, minh hoạ cho học sinh dễ nắm bắt được cú pháp và hoạt động của vòng For. Sau khi học sinh đã hiểu biết về vòng For thông qua ví dụ đơn giản trên, chúng ta cho học sinh vận dụng vào tính biểu thức có 22 dấu căn bậc hai lồng nhau để học sinh thấy vai trò cần thiết và tầm quan trọng của vòng For.

- Về phương diện dữ liệu, do tổng các số tự nhiên từ 2 đến 23 là 275 vượt ra ngoài 255 nên chúng ta phải khai biến S có kiểu Word để tránh tràn ô nhớ trong khi thực hiện phép cộng S + k. Nếu chúng ta cho tính tổng từ 1 đến 22 thì tổng này là 253 không vượt quá 255 vì thế mà cả K và S có thể khai cùng kiểu Byte, ta không có cơ hội bàn đến phương diện dữ liệu và cũng không có cơ hội nói đến quan điểm triết học "lượng đổi, chất đổi". Nếu ta cho tính tổng từ 1 đến 23 thì tổng này là 276 vượt quá 255 nhưng việc nhẩm ra 276 là không đơn giản, hơn nữa ta không có cơ hội ôn lại nhân nhẩm với 11 mà các em đã học từ trước. Ngoài ra việc tính tổng từ 1 đến 23, chúng ta cho vòng For bắt đầu từ 1 là chuyện hay gặp sau này khi làm việc với chỉ số của mảng.

Chúng ta có thể cho tính tổng từ 4 đến 23 để tổng này có 20 số hạng chia thành 10 cặp mà mỗi cặp có tổng bằng 27 vậy kết quả là 27 X 10 = 270 cũng vượt quá 255. Việc chỉ ra dãy từ 4 đến 23 có 20 số hạng không dễ bằng việc chỉ ra dãy từ 2 đến 23 cố 22 số hạng. Vì xuất phát từ 1 đến 23 có 23 số hạng là rất tự nhiên. Việc bớt đi một số chính là số 1 để còn 22 số cho dãy từ 2 đến 23 dễ nhận ra hơn là việc bớt đi ba số gồm các số 1, 2, 3 để còn 20 số cho dãy từ 4 đến 23. Hơn nữa việc nhân nhẩm với 10, 100, 1000 các em thường gặp hàng ngày. Người ta thường nói gấp mười, gấp trăm, gấp ngàn mấy ai nói gấp mười một?! Cơ hội hiếm hoi nhớ lại nhâm nhẩm với 11 chắc các em sẽ thích thú hơn là nhân nhẩm với 10 là điều mà các em đã quá quen.

Nếu chúng ta cho tính tổng từ 1 đến 100 thì cũng dễ dàng nhẩm ra tổng là 5050 như Gau xơ đã làm (1 + 100) 50, khi ông mới 6 tuổi. Nhưng vì tổng quá dễ nhận ra vượt quá 255 thì sự cảnh báo việc tràn ô nhớ sẽ không mấy ngạc nhiên với các em. Độ giật mình không cao vì các em đã chuẩn bị từ trước, không mất cảnh giác như 275.

- Về phương diện thuật toán, chúng ta đã hướng cho học sinh xây dựng hai thuật toán giải bài toán trên. Thuật toán thứ nhất là (2+23)11= 275. Thuật toán thứ hai là cộng lần lượt theo thứ tự tăng dần từ trái sang phải, tức là từ 2 đến 23. Đương nhiên thuật toán thứ nhất (2+23)11=275 là sáng tạo, thời gian thực hiện nhanh hơn, còn thuật toán thứ hai là phong cách làm việc của máy tính, để giao cho máy tính thực hiện theo vòng lặp với số lần biết trước.

- Về phương diện quá trình, việc chúng ta dùng bảng mô phỏng quá trình cộng dồn theo K như trên chính là giúp học sinh hình dung ra quá trình xảy ra bên trong bộ nhớ của máy tính khi nó thực hiện tính tổng qua vòng For. Một số dòng chúng ta chấm, chấm là cho học sinh tư duy tương tự. Việc tách dãy số thành các số độc lập là tư duy phân tích.

- Về phương diện kết quả, việc học sinh nhẩm ra kết quả 275 trước khi máy đưa ra kết quả 275 là tăng độ tin cậy vào tính đúng đắn của chương trình.

Từ chương trình tính tổng sang chương trình tính căn bậc hai là nhằm những ý tưởng sau đây:

- Rèn luyện phương diện giải quyết vấn đề. Hai bài toán khác nhau về ý nghĩa toán học nhưng qui trình tiếp cận giải quyết từng công đoạn là tương tự như nhau. đầu tiên gán cho biến lưu giữ kết quả bằng 0. Cho biến K lần lượt duyệt qua các giá trị từ giá trị ban đầu đến giá trị kết thúc. Ở bài toán tính tổng thì lấy S+K rồi gán cho biến S, ở bài toán tính căn thì lấy $\sqrt{S+K}$ rồi gán cho biến S, tức là ứng với mỗi giá trị của K cần tìm ra một biểu thức để gán cho biến S.

- Rèn kĩ thuật lập trình từ dưới lên. Học sinh biết tận dụng những chương trình đã biết, sửa chữa để có chương trình giải bài toán mới. Chủ ý cho việc này nên chúng ta đã tạo ra tình huống khai báo biến S tách riêng để đến chương trình tính căn chúng ta chỉ sửa kiểu của biến S. Việc sửa lại thân chương trình cũng là rất ít. Chúng ta chỉ thêm SQRT và hai dấu mở, đóng ngoặc tròn vào chỗ S+K. Ngoài ra chỉ thêm : 0 : 4 vào in ra số thực dấu phảy tĩnh ở đáp số.

Chính vì chúng ta muốn hấp dẫn học sinh quan tâm đến việc sửa chương trình đã có để được chương trình giải bài toán mới, nên chúng ta cố gắng thay đổi sao cho việc sửa lại là rất ít, càng ít càng tốt. Nếu ta cho giá trị của biểu thức một khác 2, giá trị của biểu thức hai khác 23 không vượt ra ngoài 255 để không phải khai lại kiểu cho K, nhưng phải sửa lại biểu thức một hoặc biểu thức hai của vòng lặp. Nếu ta cho K vượt ra khỏi kiểu Byte thì còn phải khai lại cả kiểu cho K thì thật là không nên. Cũng chủ định dùng lại chương trình tính tổng cho chương trình tính căn mà tên chương trình chúng ta đặt là Vidu, còn thông báo kết quả bài toán ta viết ‘Dap so la ‘ đó là những xâu mang ý nghĩa chung chung, thích ứng trong mọi trường hợp, không sai với mọi ngữ cảnh, để chúng ta không phải sửa lại ở một chương trình cụ thể nào.

Nếu tên chương trình tính tổng ta đặt là Tong, còn lệnh in kết quả trong thân chương trình ta viết là Writeln (‘ Tong la ‘ , S); thì sang chương trình tính căn chúng ta phải sửa lại tên chương trình là CanHai và sửa lại lệnh Writeln (‘ Ket qua la ‘, S : 0 : 4) ; thì việc sửa lại là nhiều chỗ làm giảm tính hấp dẫn dùng lại chương trình đã có đối với học sinh.

Cũng để chuẩn bị cho việc in số thực ở bài sau mà ở chỗ ‘Dap so la ‘ chúng ta đã cho vài dấu cách trước khi đóng dấu nháy cao ở chương trình tính tổng.

- Nhắc nhở học sinh phương diện sử dụng máy tính điện tử.

Việc từ chương trình tính tổng, chúng ta dán chương trình sang cửa sổ khác hoặc ghi lại tệp với tên mới rồi sửa lại bằng cách thêm, bớt, chèn, xoá để được chương trình tính căn là chỉ có thể thực hiện được với việc soạn thảo trên máy tính mà thôi.

- Bồi dưỡng quan điểm triết học.

Chúng ta chọn tính tổng từ 2 đến 23 và ta giới thiệu cách tính lần lượt từ trái qua phải là để cho học sinh thấy rằng tổng của dãy số tăng dần (lượng đổi), từ 0, 2 , 5 , 9, đến 252 thì tổng vẫn thuộc kiểu Byte, lúc thêm vào tổng số 23 thì tổng vượt ra khỏi phạm vi kiểu Byte (chất đổi).

Việc từ bài toán tính tổng các số nguyên sang bài toán tính căn bậc hai. Ở bài trước nội dung của dữ liệu lưu ở biến S là số nguyên nên biến S khai là kiểu nguyên. Sang bài sau nội dung dữ liệu lưu ở biến S là số thực vì thế mà phải khai lại

kiểu cho biến S là kiểu số thực. Tức là nội dung đã thay đổi thì hình thức cũng phải thay đổi theo cho phù hợp với nội dung mới (cặp phạm trù hình thức và nội dung).

Nói rộng ra việc sửa chương trình tính tổng để được chương trình tính căn cũng là do nội dung thay đổi (bài toán đã khác) nên hình thức (chương trình xử lí) cũng phải thay đổi theo cho phù hợp với nội dung mới.

- Bồi dưỡng cho học sinh phẩm chất của người lập trình tính cẩn thận, luân nhìn lại chương trình xem còn chỗ nào chưa hợp lí, chỗ nào còn có thể cải tiến để chương trình chạy nhanh hơn.

Rèn tính cẩn thận cho học sinh thông qua việc chúng ta nhắc họ gán cho S nhận giá trị bằng 0 trước khi viết vòng for.

Rèn ý thức cải tiến chương trình. Chúng ta đặt câu hỏi: Ở chương trình tính căn, có thể cải tiến như thế nào để rút ngắn vòng lặp xuống còn 21 lần? Nếu có học sinh phát hiện ra khởi trị cho S bằng lệnh

S :=SQRT (2) ; sau đó viết For K := 3 TO 23 DO S := SQRT (S + K) ; thì tốt. Nếu không có học sinh nào phát hiện ra điều đó thì chúng ta nhắc các em sẽ được biết qua những chương trình sau đây.

3. DownTo

Sau khi học sinh đã nắm được hoạt động của For tiến, chúng ta hình thành cho họ For lùi. Chúng ta lấy lại bài toán tính tổng các số từ 2 đến 23 đã lấy lúc ban đầu (không nên lấy ví dụ khác) S = 2 + 3 + 4 + . . . + 21 + 22 + 23

Chúng ta có nhận xét theo tính chất giao hoán của phếp cộng, tổng trên còn có thể tính theo thứ tự từ phải qua trái tức là 23 + 22 + 21 + . . . + 4 + 3 + 2. Để giao cho máy tính tổng trên theo thứ tự từ phải sang trái, Pascal có câu lệnh For lùi. Giáo viên viết cú pháp và hoạt động của For lùi như sau:

Cú pháp FOR biến := biểu thức 1 DOWNTO biểu thức 2 DO câu lệnh;

Hoạt động. Từ biểu thức 1 đến biểu thức 2 biến nhận bao nhiêu giá trị thì câu lệnh được thực hiện bấy nhiêu lần.

Sau đó chúng ta dán chương trình tính tổng ở cửa sổ thứ nhất sang cửa sổ thứ 3 và sửa lại như sau:

```
Program  ViDu ;   Uses crt;
Var      K : Byte ;  S :  Word ;
Begin
   ClrScr;   S := 23 ;
   For   K := 22  DOWNTO  2  DO    S := S + K ;
   Writeln ( ' Dap so la   ' , S ) ;
   Readln
End.
```

Việc chúng ta lấy tổng từ 2 đến 23 làm cho việc sửa chương trình từ TO sang DOWNTO là rất ít. Học sinh đã biết trước đáp số, biết định luật giao hoán trong toán học đối với dãy phép cộng, còn máy tính thì có lùi được hay không thì còn phải chờ kết quả. Khởi trị cho S bằng lệnh S := 23 là hợp lí vì đằng nào cũng khởi trị cho S thì cho S nhận giá trị ban đầu là 23 để viết vòng For lùi từ 22 đến 2 tức là giảm xuống còn 21 lần.

Khi chạy chương trình, máy đưa ra kết quả cũng là 275 làm cho học sinh tin tưởng và dễ nhận ra quá trình thực hiện vòng lặp bên trong bộ nhớ của máy tính và cũng hiểu luôn việc cải tiến để giảm số lần lặp xuống còn 21. Như vậy có nghĩa là học sinh hiểu được cú pháp và hoạt động của For lùi. Nhưng For lùi ở chương trình

trên là giải bài toán mà độ khó của nó không cao, hơn nữa là cách khác tìm ra đáp số đã biết, các em thấy nó không quan trọng và kém hứng thú. Để học sinh hứng thú và thấy tác dụng của For lùi, giáo viên cho các em xem máy giải bài toán tính S như sau:

Tính S = $\sqrt{2+\sqrt{3+...+\sqrt{22+\sqrt{23}}}}$

Bài toán này có 22 dấu căn bậc hai lồng nhau. Việc thực hiện phải lần lượt tính từ trong ra ngoài tương tự như bài toán tính căn đã giải ở trên. Theo chiều từ phải qua trái, các số xuất hiện giảm dần từ 23 đến 2. Việc dùng For lùi trong trường hợp này là thuận với qui luật tính toán và dễ hiểu chương trình sau đây:

Chúng ta dán chương trình ở cửa sổ thứ hai sang cửa sổ thứ 4, rồi sửa chương trình để có chương trình giải bài toán này. Chương trình là như sau.

```
Program  ViDu ;  Uses crt;
Var   K : Byte ;  S :  Real ;
Begin
   ClrScr;   S := SQRT ( 23 ) ;
   For   K := 22  DOWNTO  2  DO    S := SQRT(S + K) ;
   Writeln ( ' Dap so la   ' ,  S : 0 : 4 ) ;
   Readln
End.
```

Chú ý. Để câu lệnh ở thân vòng lặp For được thực hiện ít nhất là một lần thì:

Nếu For tiến thì biểu thức 1 phải không được lớn hơn biểu thức 2.

Nếu For lùi thì biểu thức 1 phải không được nhỏ hơn biểu thức 2.

Như vậy nếu biểu thức 1 bằng biểu thức 2 thì trong cả hai trường hợp câu lệnh chỉ thực hiên 1 lần là vòng lặp kết thúc.

Nếu For tiến mà biểu thức 1 lớn hơn biểu thức 2 thì câu lệnh ở thân vòng lặp không được thực hiện.

Nếu For lùi mà biểu thức 1 nhỏ hơn biểu thức 2 thì câu lệnh trong thân vòng lặp không được thực hiện.

4. Hai vòng FOR lồng nhau

Dạy vòng FOR cần phải trình bày hoạt động của 2 vòng FOR lồng nhau. Vì sau này sẽ dùng đến khi làm việc với mảng 2 chiều và trong nhiều tình huống khác.

Khi có 2 vòng FOR lồng nhau thì ứng với một giá trị của biến điều khiển vòng ngoài, máy thực hiện lần lượt với tất cả các giá trị của biến điều kiển vòng trong.

Minh hoạ 2 vòng For lồng nhau bằng việc cho học sinh quan sát kết quả khi thực hiện chương trình cho hiện lên màn hình 5 dòng đầu của bảng nhân. Chương trình như sau:

```
Program  ViDu ;   Uses crt;
   Var      d , c : Byte ;
   Begin
      clrscr;
      For   d := 1 TO  5  DO
       Begin
        For   c := 1 TO  9   DO  Write ( d : 3  ,  '.'  ,  c  ,  '='  , d*c : 2 ) ;
        Writeln
      End ;
      Readln
   End.
```

Lời bàn về phương pháp giảng dạy

Chúng ta cho in 5 dòng đầu tiên của bảng nhân mà không cho in cả bảng nhân là để cho học sinh dễ nhận ra dòng chỉ nhận 5 giá trị, còn cột nhận 9 giá trị. Nếu in cả bảng nhân thì dòng và cột đều nhận 9 giá trị làm cho việc phân biệt giá trị của dòng và giá trị của cột là không trực quan. Hơn nữa khi thực hiện chương trình, học sinh dễ nhận thấy ứng với một giá trị của dòng, máy lần lượt duyệt qua tất cả các giá trị của cột từ 1 đến 9.

Áp dụng hoạt động của hai vòng FOR lồng nhau thông qua lập trình cho máy giải bài toán 100 con trâu, 100 bó cỏ. Trâu đứng ăn 5, trâu nằm ăn 3, lụ khụ trâu già 3 con một bó. Hỏi mỗi loại có mấy con?

Chúng ta cho trâu đứng d lần lượt duyệt qua các giá trị từ 0 đến 20 (có 100 bó cỏ mà một con trâu đứng ăn 5 bó), ứng với mỗi giá trị của d chúng ta cho trâu nằm n lần lượt duyệt qua các giá trị từ 0 đến 33. Biết d, n ta sẽ tính được số trâu già là 100 – d – n.

Chương trình là như sau:

```
Program  ViDu ;   Uses crt;
Var      d , n : Byte ;
Begin
    clrscr;
    For   d := 0 TO  20  DO
      For   n := 0 TO  33   DO
          If  5 * d  + 3 * n + ( 100 – d – n ) / 3 = 100   then
            Writeln ( d , ' trau dung   ' , n , ‘ trau nam ‘ , 100 –d – n , ‘  trau gia’) ;
    Readln
End.
```

4. Sơ đồ khối

Sơ đồ hoạt động của vòng FOR tiến là như sau:

Ghi chú. BT1 là biểu thức1, BT2 là biểu thức 2