

Bài 9. LÀM VIỆC VỚI DÃY SỐ

Thời lượng: 3 tiết

1. Mục đích, yêu cầu

- Biết được khái niệm mảng một chiều;
- Biết cách khai báo mảng, nhập, in, truy cập các phần tử của mảng;
- Hiểu thuật toán tìm số lớn nhất, số nhỏ nhất của một dãy số.

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

a) Tương tự với câu lệnh điều kiện, câu lệnh lặp, vào đầu bài này cần giới thiệu một số ví dụ nhằm đưa đến nhu cầu cần có biến mảng trong ngôn ngữ lập trình.

Ví dụ ở đầu bài trong SGK dẫn đến nhu cầu biến mảng, sau khi đã phân tích sự bất tiện nếu chỉ sử dụng cách khai báo biến đã biết (khai báo biến đơn). Pascal cung cấp một công cụ hiệu quả để hỗ trợ người lập trình đó là biến mảng.

Mục 1 chỉ nêu nhu cầu của biến mảng trong ngôn ngữ lập trình và mục tiêu của các ví dụ ở mục 1 là dẫn đến nhu cầu cần có biến mảng.

Trong mục 2, cần cho HS biết cách khai báo, truy cập, nhập (gán) giá trị, viết giá trị của biến mảng ra màn hình.

b) Khai báo biến mảng

Có hai cách khai báo biến mảng

Cách 1: Khai báo trực tiếp biến mảng một chiều:

```
var <tên biến mảng>: array [<chỉ số đầu>..<chỉ số cuối>] of <kiểu dữ liệu>;
```

Cách 2: Khai báo gián tiếp biến mảng qua kiểu mảng một chiều:

```
type <tên kiểu mảng> = array [<chỉ số đầu>..<chỉ số cuối>] of <kiểu dữ liệu>;  
var <tên biến mảng>: <tên kiểu mảng>;
```

trong đó:

- <chỉ số đầu>..<chỉ số cuối> là một dãy số nguyên liên tục $n_1...n_2$ với n_1, n_2 là các hằng (hoặc biểu thức cho kết quả là số nguyên) ($n_1 \leq n_2$).
- <kiểu dữ liệu> là kiểu dữ liệu của các phần tử trong mảng.

Tuy nhiên, theo yêu cầu giảm tải, GV chỉ cần giới thiệu với HS cách 1. Kiểu chỉ số cũng chỉ cần giới thiệu thật đơn giản là dãy số nguyên dương (không nhất thiết phải giới thiệu những trường hợp còn lại). Kiểu phần tử chỉ hạn chế là số nguyên, số thực.

GV cần sử dụng một số ví dụ để luyện tập về khai báo mảng một chiều và giải thích số lượng phần tử, kiểu phần tử của từng biến mảng tương ứng với mỗi ví dụ.

c) Truy cập các phần tử của mảng

Sau khi đã cho HS luyện tập với khai báo biến mảng, có thể sử dụng các khai báo vừa thực hiện để giới thiệu về các truy cập vào biến mảng. Ví dụ, khai báo

```
var Diem: array[1..50] of real;
```

đã tạo ra một biến mảng có 50 phần tử được đánh số từ 1 đến 50. Các phần tử này được "đặt tên" như thế nào? Để "gọi đích danh" từng phần tử cụ thể Pascal sử dụng cách: *Tên biến mảng[chỉ số phần tử]*. Ví dụ, *Diem[1]* là phần tử thứ nhất; *Diem[5]* là phần tử thứ năm. Có thể thực hiện các thao tác như gán giá trị, so sánh, viết giá trị ra màn hình... với *Diem[1]*, *Diem[2]*, ..., *Diem[50]* như với biến đã học (biến đơn).

d) Nhập giá trị cho biến mảng

Để nhập giá trị cho biến mảng thì cần nhập giá trị cho từng phần tử của mảng. Giống như với việc gán giá trị cho biến đơn, có hai cách để gán giá trị cho phần tử của mảng:

Gán trực tiếp bằng lệnh gán: ví dụ: *Diem[1]:= 8, Diem[2]:= 9.5.*

Gán giá trị bằng cách nhập từ bàn phím, sử dụng lệnh *read, readln.*

Có thể viết một đoạn chương trình với 50 lệnh *readln* để thực hiện việc nhập giá trị cho 50 phần tử của mảng từ bàn phím:

```
readln(Diem[1]); readln(Diem[2]); ...; readln(Diem[50]);
```

Tuy nhiên, việc kết hợp lệnh lặp *for...do* với câu lệnh *readln* là một cách lập trình hiệu quả, thường được sử dụng để nhập dữ liệu cho mảng.

```
For i:=1 to 50 do readln(Diem[i]);
```

Tương tự như vậy, để viết giá trị của các phần tử của mảng ra màn hình người ta kết hợp giữa *for...do* với lệnh *writeln* hoặc *write*.

```
for i:=1 to 50 do writeln(Diem[i]);
```

Giả sử chỉ muốn viết ra màn hình những điểm số lớn hơn hoặc bằng 9 chẳng hạn, câu lệnh có thể được viết như sau:

```
For i:=1 to 50 do
```

```
if Diem[i] >= 9 then writeln(Diem[i]);
```

Việc đưa ra yêu cầu này có hai mục đích: thứ nhất là để HS làm quen trước với so sánh phần tử của biến mảng sẽ được sử dụng trong phần sau. Làm như vậy HS sẽ không bị ngỡ ngàng khi gặp phép so sánh này trong chương trình. Thứ hai, HS thấy được sự kết hợp giữa các câu lệnh mà cụ thể là câu lệnh *for...do* và câu lệnh *if-then* trong chương trình.

Trong các ví dụ trên, khi duyệt các phần tử của biến mảng hoàn toàn có thể sử dụng cấu trúc *while...do*, tuy nhiên ở đây cấu trúc *for...do* phù hợp hơn vì biết trước số lần lặp. Mặt khác, sử dụng cấu trúc *for...do* ở đây nói chung là dễ hiểu hơn, gần với cách nghĩ tự nhiên của HS hơn. GV cũng có thể nhắc lại về tầm quan trọng của việc lựa chọn cấu trúc điều khiển phù hợp khi lập trình.

e) Mục 3 là ví dụ về một chương trình cụ thể sử dụng biến mảng và thuật toán tìm giá trị lớn nhất, nhỏ nhất của dãy số nguyên. Trước khi giới thiệu chương trình cụ thể GV nên hướng dẫn HS tìm hiểu lại thuật toán này (đã học trong bài 5) và yêu cầu HS thảo luận, chỉnh sửa thuật toán trên để tìm ra số nhỏ nhất của dãy số. GV có thể yêu cầu HS thực hiện lại việc mô phỏng thuật toán trên một dãy số cụ thể để các em nhớ lại thuật toán. Ví dụ về mô phỏng thuật toán này đã có ở bài 5.

Giải thích về thuật toán trong chương trình tìm giá trị lớn nhất và nhỏ nhất của dãy số nguyên:

- + Đầu tiên gán giá trị số thứ nhất của dãy số cho *Max*, *Min* (ban đầu tạm thời coi số thứ nhất đồng thời là số lớn nhất và nhỏ nhất tạm thời).
- + So sánh số lớn nhất/nhỏ nhất tạm thời này với số thứ hai, nếu số thứ hai lớn hơn số lớn nhất tạm thời *Max* thì gán giá trị của số thứ hai cho *Max*; nếu số thứ hai nhỏ hơn số nhỏ nhất tạm thời *Min* thì gán giá trị của số thứ hai cho *Min*. Như vậy đến thời điểm này *Max* và *Min* là số lớn nhất và số nhỏ nhất của số thứ nhất và số thứ hai.
- + Cứ tiếp tục như vậy, đem so sánh *Max/Min* với tất cả các số còn lại, gặp số nào lớn hơn *Max* thì gán giá trị của số đó cho *Max*; gặp số nào nhỏ hơn *Min* thì gán giá trị của số đó cho *Min*. Sau khi so sánh đến số cuối cùng của dãy số thì *Max* và *Min* tương ứng chính là giá trị lớn nhất và nhỏ nhất của dãy số.

Ví dụ, mô phỏng thuật toán tìm giá trị nhỏ nhất của dãy số có thể như bảng dưới đây:

Dãy số	5	4	4	7	6	3	15	6	8
i	1	2	3	4	5	6	7	8	9
$a_i > \text{MAX}$		Sai	Sai	Đúng	Sai	Sai	Đúng	Sai	Sai
MAX	5	5	5	7	7	7	15	15	15
$a_i < \text{MIN}$		Đúng	Sai	Sai	Sai	Đúng	Sai	Sai	Sai
MIN	5	4	4	4	4	3	3	3	3

f) Để HS hiểu được máy tính làm việc như thế nào, GV có thể yêu cầu các em thực hiện như sau: Viết 10 số nguyên, mỗi số vào một mảnh giấy. Gấp 10 mảnh giấy này lại và bỏ vào một hộp A. Đặt một hộp B rỗng bên cạnh. Yêu cầu HS chuyển lần lượt đến hết từng mảnh giấy ở hộp A sang hộp B. Sau khi chuyển xong HS cho biết số lớn nhất trong các số được ghi trên các mảnh giấy. Hai bạn HS được phép xem số trên mảnh giấy khi chuyển mảnh giấy đó từ hộp A sang hộp B nhưng không ghi chép ra giấy. Mục đích của việc không cho HS ghi chép là để HS mô phỏng hoạt động của máy tính: tại thời điểm hiện tại bạn thứ nhất chỉ cần nhớ số lớn nhất, bạn thứ hai chỉ cần nhớ số nhỏ nhất, so sánh với số vừa lấy ra từ hộp A và nhớ lấy số lớn hơn hoặc nhỏ hơn theo phân công. Cứ tiếp tục như vậy đến khi hết các số trong hộp, số được mỗi bạn nhớ cuối cùng tương ứng sẽ là số lớn nhất và nhỏ nhất.

Để thực hiện công việc này cần hướng dẫn HS thực hiện mô phỏng theo giải thuật tìm dãy số lớn nhất của dãy số nguyên. Nhặt mảnh giấy đầu tiên ở hộp A, mở ra nhớ giá trị của mảnh giấy này (coi là số lớn nhất và nhỏ nhất tạm thời), gấp lại và bỏ vào hộp B. Nhặt mảnh giấy thứ hai, mở ra và mỗi bạn so sánh với số mình đang nhớ. Bạn thứ nhất nhớ giá trị lớn nhất mới nếu thấy số ghi trong mảnh giấy được lấy ra lớn hơn; bạn thứ hai nhớ giá trị nhỏ nhất mới nếu thấy số đó nhỏ hơn số mà bạn ghi nhớ. Lặp lại công việc này đến khi hết các mảnh giấy của hộp A.

GV có thể thêm, bóc các mảnh giấy để HS làm lại. Sau khi HS làm đề nghị các em mô tả lại cách các em đã thực hiện để tìm ra số lớn nhất/nhỏ nhất. Cách mà HS làm giống với cách máy tính thực hiện thuật toán ở trên. Máy

tính chỉ có thể tham chiếu đến từng số trong dãy số, máy tính không có khả năng quan sát cả dãy số vì vậy máy tính phải thực hiện theo thuật toán như trên. Máy tính thực hiện tuần tự theo đúng chỉ dẫn của con người.

Có thể cải tiến nội dung dạy học trên đây thành nhiều trò chơi khác nhau. Ví dụ, yêu cầu các em không sử dụng giấy, bút, GV lần lượt viết từng số lên bảng, rồi xoá đi luôn, HS quan sát để tìm ra số lớn nhất (hoặc nhỏ nhất). Hoặc mời một nhóm HS đứng lên phía trên lớp. Mời một em đi qua từng bạn một, khi em này đến bên bạn nào đó thì bạn này phải đưa ra một số nào đó (có thể là nói thầm hoặc viết ra một mảnh giấy). Đi hết lượt HS phải nói được bạn nào đã đưa ra số lớn nhất (hoặc nhỏ nhất). Kết quả này được kiểm chứng công khai bởi các bạn đã đưa ra các số.

Lưu ý: GV cũng có thể bắt đầu bằng chương trình tìm riêng giá trị *Max*.

Thuật toán tìm *Max* của dãy số nguyên nhập từ bàn phím như sau (Bài 5):

Bước 1. Nhập *N* và dãy A_1, \dots, A_n ;

Bước 2. $Max \leftarrow A_1$;

Bước 3. Với *i* từ 2 đến *N* thực hiện: nếu $Max < A_i$ thì $Max \leftarrow A_i$;

Bước 4. Đưa ra màn hình giá trị *Max* rồi kết thúc.

Sau khi giới thiệu xong thuật toán tìm *Max*, GV hướng dẫn HS xác định các biến, kiểu biến và viết khai báo biến; viết câu lệnh thực hiện các bước nhập *N*, nhập các phần tử của mảng, tìm *Max*, in *Max* ra màn hình. Chương trình có thể được xây dựng dần từng phần và cuối cùng có được một chương trình như dưới đây.

```
program P_Max;
Var i, N, Max: integer;
  A: array[1..100] of integer;
Begin
  {Nhập N}
  write('Hay nhập do dai của dãy số, N = '); readln(N);
  {Nhập dãy số}
  writeln('Nhập các phần tử của dãy số:');
  For i:=1 to N do
    Begin
      write('a[' , i, ']='); readln(a[i]);
    End;
  {Tìm Max}
  Max:=a[1];
```

```

for i:=2 to n do if Max<a[i] then Max:=a[i];
(Hien thi Max ra man hinh)
write('So lon nhat la Max = ',Max);
readln
End.

```

Yêu cầu HS chỉnh sửa chương trình trên để tìm giá trị nhỏ nhất của dãy số, tính tổng dãy số. Sau khi đã hiểu rõ thuật toán và chương trình tìm *Max*, *Min* có thể yêu cầu HS kết hợp tìm *Max*, *Min* trong cùng một chương trình như trong SGK.

3. Hướng dẫn trả lời câu hỏi và bài tập

Bài 1. “Có thể xem biến mảng là một biến được tạo từ nhiều biến có cùng kiểu, nhưng chỉ có một tên duy nhất”. Phát biểu này SAI. Vì từ ngay trong nội dung mệnh đề trên, đoạn “*nhiều biến có cùng kiểu, nhưng chỉ có một tên duy nhất*” là không chính xác. Trong Pascal mỗi biến đều phải có một tên khác nhau, nên không thể có nhiều biến với cùng một tên duy nhất.

Bài 2. Lợi ích chính của việc sử dụng biến mảng là rút gọn việc viết chương trình, có thể sử dụng câu lệnh lặp để thay nhiều câu lệnh. Ngoài ra chúng ta còn có thể lưu trữ và xử lý nhiều dữ liệu có nội dung liên quan đến nhau một cách hiệu quả.

Bài 3. a) Sai. Phải thay dấu phẩy bằng hai dấu chấm;
b) và c) Sai, vì chỉ số mảng phải là số nguyên;
d) Đúng.

Bài 4. Không. Giá trị nhỏ nhất và lớn nhất của chỉ số mảng phải được xác định trong phần khai báo chương trình.

Bài 5. Chương trình có thể như sau:

```

var N, i: integer;
    A: array[1..100] of real;
begin
write('Nhap so phan tu cua mang, n= '); readln(n);
for i:=1 to n do
begin
write('Nhap gia tri ',i,' cua mang, a[' ,i,']= ');
readln(a[i]);
end;
readln
end.

```