

II - HƯỚNG DẪN CHI TIẾT

§11. KIỂU MẢNG

TIẾT 1

A. Mục đích, yêu cầu

Nội dung của tiết 1 là kiểu mảng một chiều (mục 1a của §11).

Học sinh cần hiểu rõ:

- Kiểu mảng là một kiểu dữ liệu có cấu trúc, rất cần thiết và hữu ích trong nhiều chương trình.
- Mảng một chiều là một dãy hữu hạn các phần tử cùng kiểu.
- Các ngôn ngữ lập trình thông dụng cho phép người lập trình xây dựng kiểu dữ liệu mảng một chiều.

- Để mô tả mảng một chiều cần khai báo kiểu của các phần tử và cách đánh số các phần tử của nó (mỗi phần tử của nó có một chỉ số).
- Có thể tham chiếu phần tử của mảng bằng tên của mảng và chỉ số tương ứng của phần tử này.

Phần minh hoạ với Pascal, học sinh cần nắm được:

- Có hai cách để khai báo (hay định nghĩa) kiểu dữ liệu mảng một chiều:
 - Trực tiếp trong phần khai báo biến (với từ khoá *var*).
 - Đặt tên và định nghĩa cho kiểu dữ liệu mới này (với từ khoá *type*), rồi sau đó khai báo các biến thuộc kiểu mới (đã có tên).
- Để khai báo một kiểu dữ liệu mảng một chiều, người lập trình cần phải xác định những yếu tố sau:
 - Kiểu phần tử: Kiểu của các phần tử tạo nên mảng.
 - Kiểu chỉ số: Người ta thường dùng một đoạn số nguyên liên tục làm chỉ số của mảng, đoạn này được xác định khi số nguyên đầu tiên và số nguyên cuối cùng của đoạn là xác định được. Số nguyên đầu tiên trong đoạn này là chỉ số của phần tử thứ nhất trong mảng, số nguyên tiếp theo là chỉ số của phần tử thứ hai trong mảng, cứ như vậy,... giá trị cuối cùng là chỉ số của phần tử cuối cùng trong mảng. Do vậy, khi xác định kiểu chỉ số của một mảng (một chiều) người lập trình đã xác định cả kích thước của mảng (số các phần tử của mảng).

Học sinh cần có các kỹ năng:

- Nhận biết được các thành phần trong khai báo kiểu mảng một chiều.
- Nhận biết được định danh của một phần tử kiểu mảng một chiều xuất hiện trong một chương trình.
- Biết cách viết khai báo mảng đơn giản với chỉ số kiểu miền con của kiểu nguyên.

B. Những điểm cần lưu ý và gợi ý tổ chức dạy học

1. Khó khăn

- Tiết học này có khá nhiều nội dung mới đối với học sinh:
 - Với kiểu dữ liệu có cấu trúc, người lập trình có thể tự thiết kế một kiểu dữ liệu mới phức tạp hơn từ những kiểu đã có (trong Pascal có cách dùng khai báo *type* để định nghĩa một kiểu dữ liệu mới).

- Cấu trúc của một mảng một chiều.
- Khái niệm đánh số các phần tử của mảng, chỉ số, kiểu chỉ số.
- Có hai cách khai báo mảng (một chiều) trong Pascal.
- Cách truy cập đến từng phần tử của mảng.
- Trọng tâm là làm học sinh hiểu được cấu trúc của kiểu dữ liệu mảng một chiều, biết được cấu trúc này cần thiết và hữu ích trong lập trình.

2. Thuận lợi

- Dễ làm cho học sinh thấy được sự cần thiết phải có một kiểu dữ liệu mới được xây dựng như mảng một chiều và ích lợi của kiểu dữ liệu mảng.
- Dễ minh họa để học sinh thấy hình ảnh mảng và cấu trúc của mảng là quen thuộc.
- Có nhiều thuận lợi khi muốn tạo điều kiện cho học sinh tích cực và chủ động trong việc tiếp thu bài học.
- Có thể nâng cao yêu cầu và chỉ dẫn thêm cho những học sinh khá, giỏi.

3. Một số gợi ý về tổ chức dạy học

Trong ví dụ, chương trình *Nhietdo_Tuan* chỉ nhằm mục đích dẫn dắt học sinh đến tình huống cần dùng kiểu dữ liệu mảng một chiều và tiếp theo là chương trình *Nhietdo_Nngay*. Có thể bắt đầu bài học bằng việc phát biểu bài toán đầu. Sau đó, yêu cầu học sinh cho biết thuật toán và các biến của chương trình mà học sinh định viết nhằm giải quyết bài toán này. Cũng có thể giao cho học sinh bài toán đầu như một bài tập về nhà trong tiết học trước để không mất thời gian cho học sinh xây dựng và viết chương trình. Chương trình đó nên được chuẩn bị sẵn trên giấy khổ lớn (như đồ dùng dạy học) treo lên để cả lớp theo dõi hoặc chỉ yêu cầu học sinh theo dõi trong sách giáo khoa. Chú ý rằng nếu không phải kiểm tra lại nhiệt độ của từng ngày sau khi tính nhiệt độ trung bình của tuần thì chương trình không cần dùng đến bảy biến thực với mỗi biến lưu nhiệt độ của một ngày.

Khi đặt vấn đề mở rộng bài toán đó từ phạm vi một tuần sang phạm vi N ngày (chẳng hạn một năm), thì bản chất thuật toán không có gì thay đổi nhưng việc viết chương trình gặp khó khăn do cần dùng rất nhiều biến và đoạn các câu lệnh tính toán khá dài. Đặt học sinh ở tình huống này để giới thiệu kiểu dữ liệu mảng một chiều.

Có thể dùng hình vẽ mô phỏng mảng một chiều có 366 phần tử kiểu thực được đánh chỉ số từ 1 đến 366. Giải thích trên hình vẽ sao cho học sinh phân biệt được giá trị một phần tử của mảng với giá trị làm chỉ số cho chính phần tử ấy.

Nói chung, có thể không phải giới thiệu những quy định cú pháp liên quan đến kiểu mảng rồi sau đó mới vận dụng viết chương trình cho bài toán về nhiệt độ trong N ngày. Nên dùng ngay chương trình ở sách giáo khoa để học sinh làm quen với những quy tắc cú pháp mới (về mảng), giáo viên giải thích ngữ nghĩa tương ứng (có minh họa càng tốt), trên cơ sở đó khái quát hoá để giới thiệu cú pháp khai báo kiểu mảng một chiều.

Giáo viên hướng dẫn học sinh xem xét chương trình đầu tiên sử dụng kiểu mảng một chiều ở sách giáo khoa để thấy biến i lần lượt nhận các giá trị chỉ số của mảng. Tương ứng với mỗi giá trị của i , một phần tử của mảng được xác định bởi $Nhietdo[i]$, có thể thực hiện các thao tác nhập, xuất hay xử lý trên phần tử này. Như vậy, khi sử dụng một biến mảng, với câu lệnh *for-do* chương trình sẽ được viết ngắn gọn, dễ hiểu hơn.

Không nên trình bày đầy đủ về:

- Những kiểu dữ liệu nào có thể là kiểu thành phần của mảng.
- Những kiểu dữ liệu nào có thể là kiểu chỉ số.
- Kích thước cho phép của mảng.

Học sinh chỉ cần nắm được khai báo mảng một chiều với kiểu phần tử mảng là kiểu dữ liệu chuẩn, chỉ số là đoạn số nguyên liên tục.

Trong tiết học có thể dành ra ít phút để yêu cầu học sinh thực hiện vài hoạt động, chẳng hạn:

- Cho một số ví dụ về khai báo kiểu mảng một chiều hợp lệ (có thể lấy từ sách giáo khoa và thêm phần khai báo biến), yêu cầu học sinh trả lời những câu hỏi như:
 - Cho biết tên của kiểu dữ liệu mới?
 - Kiểu mảng đó có bao nhiêu phần tử?
 - Mỗi phần tử của mảng thuộc kiểu gì?
 - Qua phần khai báo biến, cho biết tên của các mảng sẽ dùng trong chương trình?...

- Nêu yêu cầu sử dụng mảng của một chương trình, đề nghị học sinh viết khai báo cho kiểu dữ liệu mảng đáp ứng được yêu cầu đó trên giấy nháp. Giáo viên kiểm tra nhanh giấy nháp của một số em để nhận xét và kết luận trên bảng. Ví dụ, một học sinh định viết một chương trình tính trung bình điểm tổng kết môn Toán các năm học từ lớp 1 đến lớp 10 của chính mình, nếu dùng một mảng để ghi nhận dữ liệu về điểm tổng kết môn Toán các năm học đó thì cần khai báo kiểu dữ liệu này ra sao?
- Có thể giới thiệu (với Pascal) cách khai báo trực tiếp kiểu mảng trong khai báo biến *var*, sau đó viết một khai báo *type* và đề nghị học sinh chuyển sang sử dụng cách khai báo kiểu trực tiếp ở khai báo *var* cho một vài biến và ngược lại (nghĩa là từ khai báo biến mà giáo viên cho học sinh chuyển sang viết khai báo kiểu với *type*). Học sinh làm trên giấy nháp, giáo viên kiểm tra nhanh giấy nháp của một vài em.

Cuối tiết học cần chốt lại về cấu trúc của kiểu dữ liệu mảng một chiều, cách khai báo và cách tham chiếu đến một phần tử của mảng. Cũng nên cho học sinh biết rằng kiểu mảng là một kiểu dữ liệu có cấu trúc được dùng nhiều trong lập trình.

TIẾT 2, 3

A. Mục đích, yêu cầu

Nội dung của tiết 2, 3 là:

Một số ví dụ về kiểu mảng một chiều (mục 1b của §11), cách sử dụng mảng một chiều trong chương trình thể hiện các thuật toán cơ bản về tìm kiếm và sắp xếp:

- Duyệt tuần tự để tìm phần tử lớn nhất của dãy số nguyên;
- Sắp xếp dãy số nguyên bằng tráo đổi;
- Tìm một số hạng trong một dãy số nguyên tăng bằng thuật toán tìm kiếm nhị phân.

Mục đích

- Củng cố và làm học sinh hiểu sâu sắc hơn những thuật toán tìm kiếm, sắp xếp cơ bản mà học sinh đã được học ở lớp 10.

- Minh họa, củng cố và nâng cao những hiểu biết của học sinh trong tiết 1 về mảng một chiều (cấu trúc, cách khai báo, cách tham chiếu một phần tử và ích lợi của kiểu dữ liệu này).
- Hình thành kỹ năng cơ bản về sử dụng kiểu mảng trong cài đặt chương trình.

Yêu cầu học sinh nắm được

- Không thể nhập hay xuất giá trị của một biến mảng một chiều như nhập hay xuất một giá trị của biến có kiểu dữ liệu chuẩn mà phải nhập hay xuất cho từng phần tử của mảng.
- Do mảng là dãy các phần tử được đánh số bởi các số nguyên liên tiếp nên việc thực hiện các thao tác nào đó (nhập, xuất hay xử lý) trên dãy các phần tử của mảng thường gắn với câu lệnh *for-do*.
- Các thao tác nhập, xuất hay xử lý mỗi phần tử của mảng phải tuân theo quy định kiểu phần tử của mảng.
- Khi khai báo kiểu mảng cần xác định kích thước của mảng. Tuy nhiên, trên thực tế có nhiều trường hợp số phần tử chứa dữ liệu thực sự có ý nghĩa đối với bài toán đang giải quyết nhỏ hơn số lượng phần tử của mảng.

B. Những điểm cần lưu ý và gợi ý tổ chức dạy học

1. Thuận lợi

- Những bài toán trong các ví dụ của bài đã được trình bày ở sách giáo khoa Tin học 10. Bởi vậy học sinh sẽ nhanh chóng hiểu được phát biểu bài toán và dễ gọi lại được ý tưởng của thuật toán. Đây là một thuận lợi cơ bản.
- Với ý nghĩa thực tế của các bài toán đặt ra, trên cơ sở học sinh đã biết thuật toán nhưng chưa trao được nhiệm vụ giải quyết những bài toán đó cho máy tính, ở bài này giáo viên dễ gây được hứng thú cho học sinh.
- Có nhiều thuận lợi khi muốn tạo điều kiện cho học sinh tích cực và chủ động trong việc tiếp thu bài học.
- Có thể nâng cao yêu cầu và chỉ dẫn thêm cho những học sinh khá, giỏi.

2. Một số gợi ý tổ chức dạy học

- Có thể yêu cầu học sinh chuẩn bị tiết học bằng cách giao nhiệm vụ cho học sinh ôn lại ba thuật toán đã học ở lớp 10 (tìm giá trị lớn nhất của một dãy số nguyên, sắp xếp bằng tráo đổi và tìm kiếm nhị phân ở §4 sách giáo khoa Tin học 10).
- Tùy theo trình độ của học sinh, không bắt buộc trình bày kỹ trên lớp đầy đủ cả ba ví dụ, nhưng tối thiểu nên trình bày hai trong ba ví dụ đó.

- Mặc dù các bài toán này đã được biết ở lớp 10, vẫn cần làm rõ phát biểu bài toán qua việc xác định input, output. Giáo viên có thể hỏi và giải thích thêm để giúp học sinh hiểu, xác định đúng yêu cầu của bài toán.
- Học sinh cần nắm được thuật toán trước khi được giáo viên hướng dẫn xem xét (hay tham gia viết) chương trình cài đặt. Giáo viên không kiểm tra hay bắt buộc học sinh diễn tả thuật toán bằng sơ đồ khối. Để biết học sinh có hiểu thuật toán hay chỉ học thuộc và phát biểu một cách máy móc, có thể yêu cầu minh họa hay mô phỏng trên một bộ dữ liệu input nhỏ (do giáo viên hay chính học sinh đưa ra). Nên tận dụng những minh họa/mô phỏng này để củng cố lại thuật toán cho cả lớp.
- Nói chung giáo viên không nên mất thời gian cho học sinh xây dựng và viết chương trình. Tuy nhiên, nên có những câu hỏi nhỏ và chi tiết giúp học sinh vận dụng hiểu biết về cấu trúc dữ liệu mảng một chiều khi thể hiện thuật toán trong ngôn ngữ lập trình, hình thành ở học sinh phương pháp "làm mịn dần" khi chuyển từ thuật toán sang chương trình. Giáo viên nên chuẩn bị sẵn diễn tả thuật toán (theo cách liệt kê các bước hay sơ đồ khối) và chương trình cài đặt trong Pascal trên giấy khổ lớn (như đồ dùng dạy học) treo lên để cả lớp theo dõi hoặc yêu cầu học sinh theo dõi trong sách giáo khoa.

Lưu ý về ví dụ 1 (tìm phần tử lớn nhất của dãy số nguyên)

- Đây là ví dụ đơn giản nhất, dễ hiểu nhất trong ba ví dụ trình bày ở mục 1.b. Tuy nhiên, cần dành thời gian thích đáng để học sinh nắm được một số điều cơ bản để tạo thuận lợi cho các ví dụ tiếp theo:
 - Dùng một mảng có kiểu phần tử là nguyên để biểu diễn một dãy hữu hạn số nguyên và cách khai báo mảng này (trong Pascal).
 - Câu lệnh *for-do* thứ nhất trong chương trình thể hiện một nhiệm vụ trong bước 1 của thuật toán, dùng để nhập các phần tử của mảng. Số phần tử thực sự của mảng do người chạy chương trình nhập vào bởi câu lệnh ngay trước câu lệnh *for-do* này.
 - Câu lệnh *for-do* thứ hai trong chương trình thể hiện vòng lặp (gồm bước 3 và 4 trong thuật toán), dùng để duyệt tuần tự từng phần tử trong mảng lọc lấy phần tử tạm thời là lớn nhất (trong các phần tử đã duyệt qua).
- Cần rèn luyện cho học sinh thuần thục cách tham chiếu đến một phần tử của mảng một chiều qua tên mảng và chỉ số của phần tử đó.

- Mỗi phần tử của mảng như vậy được xử lí như một biến kiểu nguyên.
- So với bài toán phát biểu ở lớp 10, bài toán ở đây có thêm yêu cầu cho biết chỉ số của phần tử lớn nhất và nếu có nhiều phần tử lớn nhất thì chỉ cần đưa ra một trong các chỉ số. Bởi vậy chương trình có thêm biến *csmax* để ghi nhận chỉ số của phần tử đạt giá trị lớn nhất tìm được (*Max*) trong quá trình duyệt tìm. Với chương trình trong sách giáo khoa, khi chạy chương trình, nếu dãy số nguyên nhập vào có nhiều phần tử đạt giá trị lớn nhất thì chương trình sẽ thông báo chỉ số của phần tử đầu tiên có giá trị lớn nhất. Có thể hỏi học sinh và giải thích để học sinh thấy được điều đó. Đối với học sinh khá, giỏi nên yêu cầu học sinh đề xuất thay đổi một chút trong chương trình để thông báo chỉ số của phần tử cuối cùng trong các phần tử đạt giá trị lớn nhất của dãy.

Lưu ý về ví dụ 2 (sắp xếp dãy số nguyên bằng trao đổi)

- Khi ôn lại thuật toán nên sử dụng ví dụ mô phỏng để làm rõ những điểm sau trong thuật toán, tạo điều kiện cho học sinh hiểu chương trình:
 - Khái niệm *lượt*: Sau lượt (lần) duyệt thứ nhất giá trị lớn nhất xếp đúng vị trí là ở cuối dãy. Tương tự, sau lượt thứ hai, giá trị lớn thứ hai được xếp ở vị trí sát cuối,... Sau mỗi lượt có ít nhất một số hạng đã xếp đúng vị trí. Học sinh cần hiểu cụ thể, trong thuật toán phải thực hiện bao nhiêu lượt như vậy, mỗi lượt thực hiện trên đoạn nào (từ đâu đến đâu) của dãy số? Giá trị của *j* chính là chỉ số phần tử cuối trong đoạn được duyệt của lượt. Điều này tạo điều kiện cho học sinh hiểu mục đích của câu lệnh *for j:= N downto 2 do* với biến đếm *j* chạy từ *N* về 2.
 - Mỗi lượt bao gồm việc thực hiện một số thao tác: so sánh một phần tử với phần tử đứng ngay sau nó để xử lí, bắt đầu từ phần tử đầu tiên trong dãy đến phần tử thứ *j*. Thao tác so sánh để quyết định xử lí (trao đổi hai phần tử) được lặp một số lần. Điều này tạo điều kiện cho học sinh hiểu mục đích của câu lệnh lặp mà chương trình dùng để thể hiện mỗi lượt.

```

for i:=1 to j-1 do
  if A[i]> A[i+1] then
    begin (* Trao doi A[i] va A[i+1] *)
      t:= A[i];
      A[i]:= A[i+1];
      A[i+1]:=t;
    end;

```


Giá trị biến đếm i chính là chỉ số phần tử được lấy so sánh với phần tử kế sau nó trong dãy số. Học sinh cần hiểu vì sao trong cấu trúc lặp này, i chỉ lấy đến $j - 1$ chứ không phải là đến j .

- Nên bám sát thuật toán để giải thích (hoặc xây dựng) những đoạn câu lệnh tương ứng trong chương trình.
- Có thể gợi ý cho học sinh khá giỏi viết lại chương trình khi dùng cấu trúc lặp *while-do* thay cho cấu trúc *for-do*.
- Có thể yêu cầu học sinh đề xuất thay đổi chương trình để sắp xếp dãy số thành dãy không tăng.

Lưu ý về ví dụ 3 (thuật toán tìm kiếm nhị phân)

- Khi ôn lại thuật toán, nên củng cố và nhấn mạnh ý tưởng của thuật toán là *thu hẹp phạm vi tìm kiếm* sau mỗi lần so sánh khoá với số hạng được chọn. Cũng nên sử dụng ví dụ mô phỏng để làm rõ một số điểm trong thuật toán, tạo điều kiện cho học sinh hiểu chương trình:
 - Trong thuật toán, việc tìm kiếm thực chất là lặp một số lần (chưa xác định được trước) các thao tác sau: chọn số hạng ở "giữa" dãy, so sánh số hạng đó với k , căn cứ vào kết quả so sánh này để hoặc kết luận đã tìm thấy (trường hợp xảy ra *bằng*) hoặc thu hẹp phạm vi tìm kiếm (trường hợp *không bằng*).
 - Khi nào thì quá trình lặp nói trên dừng lại? Quá trình lặp đó cần dừng lại khi một trong hai sự kiện sau xảy ra: đã tìm thấy hoặc không gian tìm kiếm đã trở nên bằng rỗng (nghĩa là không còn đoạn nào của dãy cho ta hi vọng chứa phần tử cần tìm).
 - Phạm vi tìm kiếm trên dãy là một đoạn được xác định bởi các biến nguyên *Dau* và *Cuoi*, tương ứng cho biết bắt đầu từ phần tử có chỉ số *Dau* của dãy cho đến phần tử có chỉ số *Cuoi* của dãy. Từ đó, dẫn dắt để học sinh hiểu được công thức xác định phần tử ở "giữa" phạm vi tìm kiếm và công thức xác định lại giá trị cho biến *Dau* hay *Cuoi* trong mỗi trường hợp thu hẹp phạm vi tìm kiếm.
- Từ thuật toán chuyển sang chương trình, nên dẫn dắt học sinh theo cách "làm mịn dần" với một vài điểm cần lưu ý:
 - Phần cơ bản của chương trình sẽ gồm một cấu trúc lặp (chưa xác định trước được số lần lặp).

- Cần ghi nhận được sự kiện *tìm thấy*, có thể dùng một biến logic *Tim_Thay* để ghi nhận. Khi chưa tìm kiếm, tất nhiên phải khởi tạo biến này là *false*. Khi tìm thấy sẽ đổi giá trị của biến *Tim_Thay* thành *true*. Điều này làm dễ dàng xác định điều kiện lặp.
- Điều kiện của lặp là gì? Thể hiện sự kiện chưa tìm thấy hoặc không gian tìm kiếm chưa rộng bằng biểu thức logic nào?
- Khi kết thúc lặp, giá trị của biến logic *Tim_Thay* cho biết có tìm thấy hay không, bởi vậy sau cấu trúc lặp, dùng câu lệnh rẽ nhánh theo giá trị của *Tim_Thay* để thông báo kết quả.
- Trường hợp tìm thấy, cần đưa ra kết quả chi tiết hơn, cần thông báo chỉ số của phần tử có giá trị là *k*. Khi tìm thấy, sự kiện này được ghi nhận ngay (ngay sau khi so sánh phần tử *giữa* được chọn với *k*), rồi điều kiện lặp được kiểm tra và quá trình lặp dừng lại. Bởi vậy, lúc đó biến *Giua* cho biết chỉ số của phần tử cần tìm.

Cuối tiết học nên nhắc lại rằng kiểu mảng là một kiểu dữ liệu có cấu trúc được dùng nhiều trong lập trình, những bài toán tìm kiếm, sắp xếp dãy các phần tử là những bài toán thường gặp và có ý nghĩa quan trọng. Chốt lại cho học sinh về cấu trúc của kiểu dữ liệu mảng một chiều, cách khai báo và cách tham chiếu đến một phần tử của mảng.

TIẾT 4

A. Mục đích, yêu cầu

Nội dung của tiết 4 là kiểu mảng hai chiều (mục 2 của §11).

Học sinh cần biết:

- Có thể hình dung một mảng hai chiều như một bảng có các dòng và các cột. Các dòng và các cột đều được đánh số bằng các số nguyên liên tiếp. Hai đoạn số nguyên dùng để đánh số cho các dòng và các cột có thể khác nhau. Mỗi phần tử của mảng hai chiều được hình dung như giá trị của một ô trong bảng, mỗi ô có chỉ số dòng và chỉ số cột.
- Giống như mảng một chiều, có thể tham chiếu (hay thao tác) trên mỗi phần tử của mảng hai chiều, nhưng khác ở chỗ mỗi phần tử của mảng hai chiều có hai chỉ số (chứ không phải một chỉ số như ở mảng một chiều) và do vậy nó được xác định bởi tên mảng và hai chỉ số của nó.

- Mảng hai chiều là mảng một chiều mà mỗi phần tử của nó lại là một mảng một chiều.

Phần minh họa với Pascal, học sinh cần nắm được:

- Cách khai báo kiểu dữ liệu mảng hai chiều:

+ Khai báo gián tiếp biến mảng qua kiểu mảng hai chiều:

```
type <tên kiểu mảng>=array[kiểu chỉ số dòng, kiểu chỉ số cột] of <kiểu phần tử>;
```

```
var <biến mảng>: <tên kiểu mảng>;
```

+ Khai báo trực tiếp biến mảng hai chiều:

```
var <biến mảng>: array[kiểu chỉ số dòng, kiểu chỉ số cột] of <kiểu phần tử>;
```

- Giống như khi khai báo kiểu dữ liệu mảng một chiều, người lập trình cần phải xác định kiểu của các phần tử tạo nên mảng và kiểu chỉ số. Cách xác định kiểu chỉ số vẫn như đã biết ở kiểu mảng một chiều, chỉ khác là ở mảng hai chiều cần xác định hai chỉ số, hai chỉ số đó độc lập với nhau.
- Giống như ở mảng một chiều, các thao tác nhập, xuất hay xử lý mỗi phần tử của mảng phải tuân theo quy định kiểu phần tử của mảng.
- Việc thực hiện thao tác nào đó (nhập, xuất hay xử lý) lần lượt trên các phần tử của mảng hai chiều thường gắn với hai câu lệnh *for-do* lồng nhau.

Học sinh cần có các kĩ năng:

- Nhận biết được các thành phần trong khai báo kiểu mảng hai chiều.
- Nhận biết được định danh của một phần tử mảng hai chiều xuất hiện trong một chương trình. Cách tham chiếu đến một phần tử của mảng hai chiều.
- Viết được khai báo mảng hai chiều với các chỉ số thuộc kiểu nguyên.

B. Những điểm cần lưu ý và gợi ý tổ chức dạy học

Trên cơ sở học sinh đã học về mảng một chiều, việc cung cấp kiến thức cho học sinh về mảng hai chiều là thuận lợi. So với tiết đầu tiên trong chương (dạy về mảng một chiều), ở tiết này học sinh gặp ít nội dung mới hơn. Giáo viên nên củng cố những kiến thức đã có về mảng một chiều trong tiết học này đồng thời tận dụng những hiểu biết đó để xây dựng kiến thức mới về mảng hai chiều cho học sinh.

Có thể bắt đầu tiết học bằng việc kiểm tra hiểu biết của học sinh về mảng một chiều hoặc chữa nhanh một bài tập cơ bản (đã giao cho học sinh làm ở nhà)

về mảng một chiều. Câu hỏi kiểm tra có thể lấy từ các câu hỏi từ 1 đến 4 ở sách giáo khoa, giáo viên có thể hỏi thêm để kiểm tra học sinh về kỹ năng khai báo mảng và tham chiếu phần tử thuộc mảng. Phần kiểm tra đầu tiết nhằm mục đích nhắc lại những kiến thức cơ bản về mảng một chiều, dẫn dắt học sinh sang kiểu mảng hai chiều.

Nên dùng hình vẽ (trong sách giáo khoa) một bảng có chỉ số theo dòng và cột để minh họa cho một mảng hai chiều. Giải thích cú pháp của câu lệnh khai báo kiểu mảng hai chiều, dùng hình vẽ này để minh họa. Lưu ý học sinh rằng hai đoạn số nguyên dùng đánh chỉ số cho dòng và cột của bảng có thể khác nhau.

Về bản chất kiểu của chỉ số là kiểu miền con, tuy nhiên để cho đơn giản, sách giáo khoa không giới thiệu tường minh khái niệm kiểu miền con, mà chỉ giới hạn ở đoạn số nguyên liên tục.

Bởi vậy, một mặt khi trình bày giáo viên nên tránh các ví dụ phải sử dụng chỉ số khác với đoạn số nguyên, mặt khác nếu có học sinh nào sử dụng kiểu dữ liệu này thì giáo viên vẫn có thể khuyến khích.

Lưu ý về ví dụ 1: Có thể gợi ý và hướng dẫn để học sinh tham gia viết được chương trình cho bài toán đặt ra.

Lưu ý về ví dụ 2:

- Để lần lượt có được giá trị các phần tử tạo nên mảng hai chiều B , thường dùng hai vòng lặp *for-do* lồng nhau. Câu lệnh viết trong hai vòng lặp này làm việc với một phần tử của mảng có hai chỉ số tương ứng là hai biến điều khiển của hai vòng *for-do*. Có thể yêu cầu học sinh cho biết câu lệnh cần viết trong các tình huống:
 - Nhập từ bàn phím giá trị phần tử này.
 - Mong đợi ở học sinh câu trả lời: (giáo viên sẽ chỉnh sửa, kết luận)

```
write('B[' , i , ', ' , j , ']= ');  
readln(B[i, j]);
```
 - Lệnh tính giá trị phần tử này theo giá trị của hai chỉ số (chẳng hạn như ở ví dụ 1).
- Muốn đưa ra màn hình tất cả các phần tử của mảng có giá trị lớn hơn một giá trị nào đó cần phải duyệt qua tất cả các phần tử của mảng, có thể dùng một số câu hỏi gợi ý để học sinh tham gia viết đoạn chương trình như vậy.

Chẳng hạn, một phương án gợi ý cho học sinh như sau: "Chương trình ở ví dụ 1 cũng đã phải duyệt qua tất cả các phần tử của một mảng hai chiều để in ra tất cả các phần tử, trong trường hợp này không phải phần tử nào cũng được đưa ra mà chỉ đưa ra những phần tử thoả mãn điều kiện lớn hơn giá trị của biến k . Vậy có thể sửa đổi đoạn chương trình trong ví dụ 1 như thế nào để đáp ứng được yêu cầu đặt ra?"

Nếu còn thời gian, giáo viên lưu ý học sinh rằng có thể dùng khai báo kiểu dữ liệu mảng hai chiều với từ khoá *type*. Giáo viên viết lại các khai báo trong ví dụ (hoặc trong chương trình của ví dụ) sang dạng này.

Cuối tiết học, cần chốt lại những hiểu biết chung nhất về kiểu dữ liệu mảng:

- Kiểu mảng một chiều thường được dùng trong những chương trình cần tổ chức dữ liệu như một dãy các phần tử cùng kiểu để giải quyết được bài toán đặt ra.
- Khi cần tổ chức dữ liệu có cấu trúc bảng, người ta nghĩ đến việc dùng mảng hai chiều.
- Mỗi biến mảng gồm nhiều phần tử và mỗi phần tử có chỉ số: Mảng một chiều có một chỉ số, mảng hai chiều có hai chỉ số. Mỗi phần tử của mảng một chiều (hay hai chiều) được tham chiếu qua tên mảng và một (hay hai) chỉ số.

C. Kiến thức bổ sung

- Kiểu của chỉ số có thể là kiểu miền con của các kiểu nguyên (byte, integer, word, longint), kiểu kí tự (char), kiểu lôgic (boolean). Trên dữ liệu thuộc những kiểu này áp dụng được các hàm chuẩn *pred(x)*, *succ(x)*, *ord(x)* và biến điều khiển trong các vòng lặp *for-do* phải thuộc một trong những kiểu dữ liệu này. Do kiểu lôgic (boolean) chỉ có hai giá trị (*true* và *false*) nên ít dùng kiểu lôgic làm kiểu chỉ số khi khai báo kiểu mảng. Pascal còn cho phép người dùng định nghĩa kiểu liệt kê, kiểu liệt kê cũng có thể là kiểu chỉ số cho khai báo mảng. Tuy nhiên, trong sách giáo khoa không giới thiệu về kiểu liệt kê do người dùng định nghĩa, nên không yêu cầu học sinh biết kiểu chỉ số có thể là kiểu liệt kê. Những kiến thức liên quan đến kiểu liệt kê do người dùng định nghĩa chỉ dùng để mở rộng kiến thức hay giải đáp thắc mắc cho học sinh giỏi.
- Có thể khai báo mảng với chỉ số kiểu kí tự. Ví dụ, các khai báo sau là hợp lệ:

```
var M: array['A'..'Z'] of integer;
```

hoặc

```
type Mang = array[1..100, 'A'..'H'] of real;
```

- Khi khai báo kích thước của mảng cần chú ý đến dung lượng cho phép của bộ nhớ dành cho chương trình. Có những khai báo không sai cú pháp nhưng vượt quá kích thước cho phép, chương trình dịch sẽ báo lỗi "Structure too large", ví dụ:

```
var M, N: array[1..10000] of longint;  
{chiếm:  $2 \times 10000 \times 4$  byte > 64 KB}
```

hoặc

```
var T: array[1..1000, 1..100] of integer;  
{vượt 64 KB}
```

Chú ý: Vùng nhớ cấp phát cho dữ liệu của một môđun chương trình Pascal tối đa là 64 KB.