

§11. KIỂU MẢNG

Chúng ta chỉ xét hai kiểu mảng thông dụng với nhiều ngôn ngữ lập trình là kiểu mảng một chiều và kiểu mảng hai chiều.

1. Kiểu mảng một chiều

Mảng một chiều là dãy hữu hạn các phần tử cùng kiểu. Mảng được đặt tên và mỗi phần tử của nó có một chỉ số. Để mô tả mảng một chiều cần xác định kiểu của các phần tử và cách đánh số các phần tử của nó.

Để người lập trình có thể xây dựng và sử dụng kiểu mảng một chiều, các ngôn ngữ lập trình có quy tắc, cách thức cho phép xác định:

- Tên kiểu mảng một chiều;
- Số lượng phần tử;
- Kiểu dữ liệu của phần tử;
- Cách khai báo biến mảng;
- Cách tham chiếu đến phần tử.

Ví dụ, xét bài toán nhập vào nhiệt độ (trung bình) của mỗi ngày trong tuần, tính và đưa ra màn hình nhiệt độ trung bình của tuần và số lượng ngày trong tuần có nhiệt độ cao hơn nhiệt độ trung bình của tuần.

Ta có thể dùng bảy biến thực để lưu trữ nhiệt độ của các ngày trong tuần. Chương trình giải bài toán có thể được viết bằng Pascal như sau:

```
program Nhietdo_Tuan;
var t1,t2,t3,t4,t5,t6,t7,tb: real;
    dem: integer;
begin
    writeln('Nhap vao nhiet do cua 7 ngay: ');
    readln(t1,t2,t3,t4,t5,t6,t7);
    tb:= (t1+t2+t3+t4+t5+t6+t7)/7;
    dem:= 0;
    if t1>tb then dem:= dem+1;
```

```

    if t2>tb then dem:= dem+1;
    if t3>tb then dem:= dem+1;
    if t4>tb then dem:= dem+1;
    if t5>tb then dem:= dem+1;
    if t6>tb then dem:= dem+1;
    if t7>tb then dem:= dem+1;
    writeln('Nhiệt độ trung bình tuần: ',tb:4:2);
    writeln('Số ngày nhiệt độ cao hơn trung bình: ',dem);
    readln
end.

```

Khi cần giải bài toán trên với N ngày (N khá lớn) thì cách làm tương tự không những đòi hỏi một khối lượng khai báo khá lớn, mà đoạn chương trình tính toán cũng khá dài.

Để giải quyết vấn đề đó, ta sử dụng kiểu dữ liệu mảng một chiều để mô tả dữ liệu. Chương trình giải bài toán tổng quát với N ngày trong Pascal có thể như sau:

```

program Nhiệtđộ_Ngày;
const Max = 366;      {giả thiết N lớn nhất là 366}
type Kmaingl= array[1..Max] of real;
var Nhiệtđộ: Kmaingl;
    dem, i, N: integer;
    Tong, Trung_bình: real;
begin
    write('Nhập số ngày: '); readln(N);
    Tong:= 0;
    for i:= 1 to N do
        begin
            write('Nhập nhiệt độ ngày ',i, ': ');
            readln(Nhiệtđộ[i]);
            Tong:= Tong + Nhiệtđộ[i];
        end;
    dem:= 0;
    Trung_bình:= Tong/N;
    for i:= 1 to N do
        if Nhiệtđộ[i]>Trung_bình then dem:= dem+1;
    writeln('Nhiệt độ trung bình',N,' ngày: ',Trung_bình:8:3);
    writeln('Số ngày nhiệt độ cao hơn trung bình: ',dem);
    readln
end.

```

Trong chương trình này đã khai báo biến mảng một chiều (thông qua khai báo kiểu mảng) như sau:

```

Type Kmgangl = array[1..Max] of real;
Var Nhietdo: Kmgangl;

```

Khai báo *kiểu mảng* một chiều gồm *Max* số thực.
Khai báo *biến mảng* *Nhietdo* qua kiểu mảng.

a) Khai báo

Tổng quát, khai báo biến mảng một chiều có dạng:

– Cách 1. Khai báo *trực tiếp* biến mảng một chiều:

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

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

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

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

trong đó:

- *Kiểu chỉ số* thường là một đoạn số nguyên liên tục có dạng $n1..n2$ với $n1, n2$ là các hằng hoặc biểu thức nguyên xác định chỉ số đầu và chỉ số cuối ($n1 \leq n2$);
- *Kiểu phần tử* là kiểu của các phần tử mảng.

Ví dụ. Các khai báo kiểu mảng một chiều sau đây là hợp lệ:

type

```

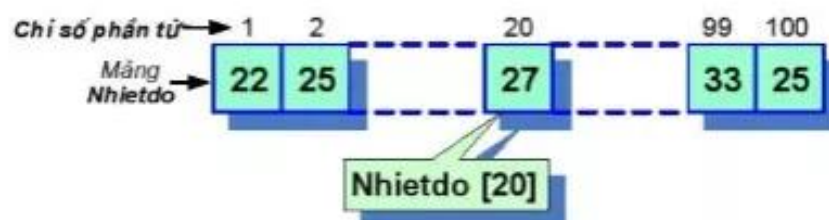
ArrayReal = array[-100..200] of real;
ArrayBoolean = array[-n+1..n+1] of boolean;
ArrayInt = array[-100..0] of integer;

```

trong đó n là hằng nguyên.

Tham chiếu tới phần tử của mảng một chiều được xác định bởi tên mảng cùng với chỉ số, được viết trong cặp ngoặc [và].

Ví dụ, tham chiếu tới nhiệt độ của ngày thứ 20, trong chương trình trên, được viết là *Nhietdo*[20].



Hình 10. Minh họa mảng một chiều

b) Một số ví dụ

Ta xét chương trình có sử dụng mảng một chiều cài đặt một số thuật toán giải những bài toán tìm kiếm và sắp xếp.

Ví dụ 1. *Tìm phần tử lớn nhất của dãy số nguyên*

Input: Số nguyên dương N ($N \leq 250$) và dãy N số nguyên dương A_1, A_2, \dots, A_N , mỗi số đều không vượt quá 500.

Output: Chỉ số và giá trị của phần tử lớn nhất trong dãy số đã cho (nếu có nhiều phần tử lớn nhất chỉ cần đưa ra một trong số chúng).

Bước 1. Nhập N và dãy A_1, \dots, A_N ;
Bước 2. $\text{Max} \leftarrow A_1, i \leftarrow 2$;
Bước 3. Nếu $i > N$ thì đưa ra giá trị Max rồi kết thúc;
Bước 4.
 Bước 4.1. Nếu $A_i > \text{Max}$ thì $\text{Max} \leftarrow A_i$;
 Bước 4.2. $i \leftarrow i + 1$ rồi quay lại bước 3;

Hình 11. Thuật toán tìm phần tử lớn nhất của dãy số

Chương trình dưới đây thực hiện việc duyệt tuần tự các phần tử để tìm ra số lớn nhất của dãy số nguyên.

```
program TimMax;
uses crt;
const
  Nmax = 250;
type
  ArrInt = array[1..Nmax] of integer;
var
  N, i, Max, csmax: integer;
  A: ArrInt;
begin
  clrscr;
  write('Nhap so luong phan tu cua day so, N = ');
  readln(N);
  for i:=1 to N do
    begin
      write('Phan tu thu ', i, ' = ');
      readln(A[i]);
    end;
```

```

Max:= A[1]; csmax:=1;
for i:=2 to N do
    if A[i]> Max then
        begin
            Max:= A[i];
            csmax:= i;
        end;
writeln('Gia tri cua phan tu Max: ', Max);
writeln('Chi so cua phan tu Max: ', csmax);
readln
end.

```

Ví dụ 2. Sắp xếp dãy số nguyên bằng thuật toán đảo ngược

Input: Số nguyên dương N ($N \leq 250$) và dãy A gồm N số nguyên dương A_1, A_2, \dots, A_N , mỗi số đều không vượt quá 500.

Output: Dãy số A đã được sắp xếp thành dãy không giảm.

Để giải bài toán này, ta sẽ sử dụng thuật toán đảo ngược như mô tả trong sách giáo khoa Tin học 10.

```

(* Chương trình giải bài toán sắp xếp dãy số *)
program sapxep;
uses crt;
const Nmax = 250;
type
    ArrInt = array[1..Nmax] of integer;
var
    N,i,j,t: integer;
    A: ArrInt;
begin
    clrscr;
    write('Nhap so luong phan tu cua day so, N = ');
    readln(N);
    for i:=1 to N do (* Nhap cac phan tu *)
        begin
            write('Phan tu thu ',i,' = ');
            readln(A[i]);
        end;
    for j:=N downto 2 do
        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;
writeln('Day so duoc sap xep la: ');
for i:=1 to N do write(A[i]: 4);
readln
end.

```

Ví dụ 3. Tìm kiếm nhị phân

Input: Dãy A là dãy tăng gồm N ($N \leq 250$) số nguyên dương A_1, A_2, \dots, A_N và số nguyên k .

Output: Chỉ số i mà $A_i = k$ hoặc thông báo "*Không tìm thấy*" nếu không có số hạng nào của dãy A có giá trị bằng k .

Để giải bài toán này, chúng ta sẽ sử dụng thuật toán tìm kiếm nhị phân được trình bày trong sách giáo khoa Tin học 10.

Bước 1. Nhập N , các số hạng A_1, A_2, \dots, A_N và khoá k ;
Bước 2. $Dau \leftarrow 1, Cuoi \leftarrow N$;
Bước 3. $Giua \leftarrow \left\lfloor \frac{Dau + Cuoi}{2} \right\rfloor$;
Bước 4. Nếu $A_{Giua} = k$ thì thông báo chỉ số $Giua$, rồi kết thúc;
Bước 5. Nếu $A_{Giua} > k$ thì đặt $Cuoi = Giua - 1$ rồi chuyển đến bước 7;
Bước 6. $Dau \leftarrow Giua + 1$;
Bước 7. Nếu $Dau > Cuoi$ thì thông báo dãy A không có số hạng có giá trị bằng k , rồi kết thúc;
Bước 8. Quay lại bước 3.

Hình 12. Thuật toán tìm kiếm nhị phân

```

(* Chương trình cài đặt thuật toán tìm kiếm nhị phân *)
program TK_nhiphan;
uses crt;
const
    Nmax = 250;
type
    ArrInt = array[1..Nmax] of integer;
var
    N, i, k: integer;
    Dau, Cuoi, Giua: integer;
    A: ArrInt;
    Tim_Thay: boolean;

```

```

begin
  clrscr;
  write('Nhap so luong phan tu cua day so, N = ');
  readln(N);
  writeln('Nhap cac phan tu cua day so tang: ');
  for i:=1 to N do
    begin
      write('Phan tu thu ',i,' = ');
      readln(A[i]);
    end;
  write('Nhap gia tri k = ');
  readln(k);
  Dau:= 1; Cuoi:=N; Tim_thay:= false;
  while (Dau <= Cuoi) and not (Tim_Thay) do
    begin
      Giua:= (Dau+Cuoi) div 2;
      if A[Giua] = k then
        Tim_thay:= true
      else
        if A[Giua]>k then Cuoi:= Giua-1
        else Dau:= Giua+1;
    end;
  if Tim_thay then writeln('Chi so tim duoc la: ', Giua)
  else writeln('Khong tim thay');
  readln
end.

```

2. Kiểu mảng hai chiều

Xét bài toán tính và đưa ra màn hình bảng nhân.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90

Bảng nhân

Ta thấy bảng nhân có dạng bảng gồm các giá trị cùng kiểu. Ta có thể biểu diễn bảng nhân bằng kiểu dữ liệu mảng hai chiều.

Mảng hai chiều là bảng các phần tử cùng kiểu.

Nhận xét rằng mỗi hàng của mảng hai chiều có cấu trúc như một mảng một chiều cùng kích thước. Nếu coi mỗi hàng của mảng hai chiều là một phần tử thì ta có thể nói mảng hai chiều là mảng một chiều mà mỗi phần tử là mảng một chiều.

Như vậy, ta cũng có thể mô tả dữ liệu của bảng nhân là kiểu mảng một chiều gồm 9 phần tử, mỗi phần tử lại là mảng một chiều có 10 phần tử, mỗi phần tử là một số nguyên.

Tương tự như với kiểu mảng một chiều, với kiểu mảng hai chiều, các ngôn ngữ lập trình cũng có các quy tắc, cách thức cho phép xác định:

- Tên kiểu mảng hai chiều;
- Số lượng phần tử của mỗi chiều;
- Kiểu dữ liệu của phần tử;
- Cách khai báo biến;
- Cách tham chiếu đến phần tử.

Ví dụ, biến mảng hai chiều *B* lưu trữ bảng nhân có thể được khai báo trong Pascal như sau:

```
var B: array[1..9] of array[1..10] of integer;
```

hoặc có thể khai báo ngắn gọn:

```
var B: array[1..9,1..10] of integer;
```

a) *Khai báo*

Tổng quát, khai báo biến mảng hai chiều trong Pascal như sau:

– *Cách 1*. Khai báo trực tiếp biến mảng hai chiều:

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

– *Cách 2*. Khai báo gián tiếp biến mảng qua kiểu mảng hai chiều:

```
type <tên kiểu mảng> = array[kiểu chỉ số hàng, kiểu chỉ số cột]  
                                of <kiểu phần tử>;  
  
var <tên biến mảng>: <tên kiểu mảng>;
```

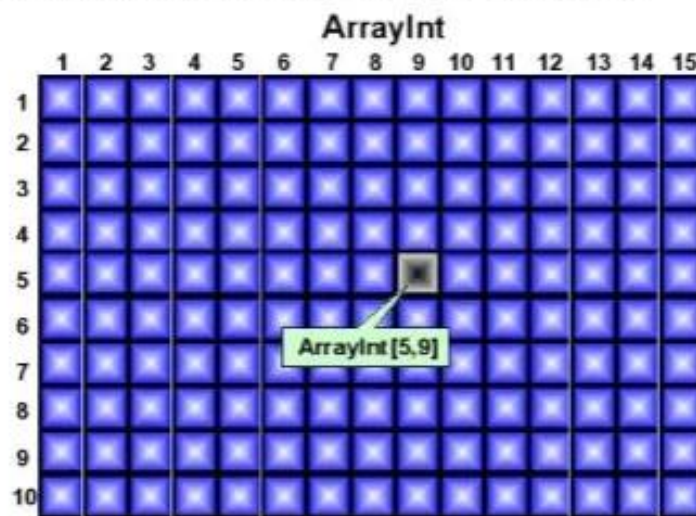

Ví dụ. Các khai báo sau đây là hợp lệ:

```
type
  ArrayReal = array[-100..200,100..200] of real;
  ArrayBoolean = array[-n+1..n+1,n..2*n] of boolean;
var
  ArrayInt: array[1..10,1..15] of integer;
  ArrayLong: array[0..3*(n+1),0..n] of longint;
```

trong đó n là hằng nguyên.

Tham chiếu tới phần tử của mảng hai chiều được xác định bởi tên mảng cùng với hai chỉ số được phân cách bởi dấu phẩy và viết trong cặp ngoặc [và].

Ví dụ. Tham chiếu tới phần tử ở hàng thứ 5, cột thứ 9 của biến mảng *ArrayInt* khai báo trong ví dụ trên được viết: *ArrayInt* [5, 9].



Hình 13. Minh họa mảng hai chiều

b) Một số ví dụ

Ví dụ 1. Chương trình sau tính và đưa ra màn hình bảng nhân.

```
program Bang_nhan;
uses crt;
var
  B: array[1..9,1..10] of integer;
  {B: bien mang hai chieu luu bang nhan}
  i, j: integer;
begin
  clrscr;
  for i:=1 to 9 do
    for j:= 1 to 10 do
      B[i,j]:= i*j;
  for i:=1 to 9 do
```

```

        begin
            for j:=1 to 10 do write(B[i,j]:4);
            writeln;
        end;
    readln
end.

```

Ví dụ 2. Chương trình sau nhập vào từ bàn phím các phần tử của mảng hai chiều B gồm 5 hàng, 7 cột với các phần tử là các số nguyên và một số nguyên k . Sau đó, đưa ra màn hình các phần tử của mảng có giá trị nhỏ hơn k .

```

program MangHaiChieu;
uses crt;
var b: array[1..5, 1..7] of integer;
    d, i, j, k: integer;
begin
    clrscr;
    writeln('Nhap cac phan tu cua mang theo dong: ');
    for i:= 1 to 5 do
        begin
            for j:= 1 to 7 do read(b[i,j]);
            writeln;
        end;
    write('Nhap vao gia tri k = '); readln(k);
    d:= 0;
    writeln('DS cac phan tu mang nho hon ',k,':');
    for i:= 1 to 5 do
        for j:= 1 to 7 do
            if b[i,j] < k then begin
                write(b[i,j], ' ');
                d:= d+1;
            end;
    if d = 0 then writeln('Khong co phan tu nao nho hon ',k);
    readln
end.

```

Chú ý:

- Các biến mảng thường gồm số lượng lớn các phần tử nên cần lưu ý phạm vi sử dụng chúng để khai báo kích thước và kiểu dữ liệu sao cho tiết kiệm bộ nhớ.
- Ngoài hai kiểu mảng một chiều và hai chiều, còn có kiểu mảng nhiều chiều.