CHƯƠNG IV. KIỂU DỮ LIỆU CÓ CẤU TRÚC

BÀI 11 : KIỂU MẢNG

Có hai kiểu mảng thông dụng với nhiều NNLT 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.

1. ***Khai báo***

* ***Cách 1*** : khai báo trực tiếp biến mảng một chiều. **Var**< *tên biến mảng>* ***:array****[kiểu chỉ so]* **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** xác định chỉ số đầu và chỉ số cuối (với nl, n2 là các hằng hoặc biểu thức nguyên và n1<n2).
* *Kiểu phần tử*: là kiểu của các phần tử mảng.

1. ***Tham chiếu***

* 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 phần tử thứ 20 của biến mảng NhietDo ở trên được viết là **NhietDo[20]**

1. ***Một số ví dụ***

* ***Ví dụ 1*** : Tìm phần tử 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 luongphan 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**.

**2. Kiểu mảng hai chiều** *(tham khảo thêm SGK)*

**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.

xxxxx

BÀI 12 : KIỂU XÂU

* *Xâu* là dãy kí tự trong bộ mã ASCII, mỗi kí tự được gọi là một phần tử của xâu.
* Số lượng kí tự trong một xâu gọi là độ dài của xâu.
* Xâu có độ dài bằng 0 gọi là xâu rỗng.
* Có thể xem xâu là mảng một chiều mà mỗi phần tử là một kí tự.
* Các kí tự của xâu được đánh số thứ tự, thường bắt đầu từ 1.
* Tham chiếu tới phần tử của xâu được xác định bởi tên biến xâu và chỉ số đặt trong cặp ngoặc [ và ].

1. **Khai báo**

* Để khai báo kiểu dữ liệu xâu ta sử dụng tên dành riêng ***string****,* tiếp theo là độ dài lớn nhất của xâu (không vượt quá 255) được ghi trong cặp ngoặc [ và ].
* Biến kiểu xâu khai báo như sau:

**Var**< *tên bìến>****:* string** *[độ dài lớn nhất của xâu];*

* Khi khai báo biến xâu có thể bỏ qua phần khai báo độ dài, lúc đó độ dài lớn nhất của xâu sẽ nhận giá trị ngầm định là 255

1. **Các thao tác xử lí xâu**
2. **Phép ghép xâu**

* kí hiệu là d ấu (+), được sử dụng để ghép nhiều xâu thành một.

*Ví dụ* : phép ghép xâu

'Ha' + ' Noi' + ' - ' + 'Viet Nam'

cho xâu kết quả là 'Ha Noi - Viet Nam'

* Có thể thực hiện phép ghép xâu đối với các *hằng* và *biến xâu Ví dụ:*

biến xâu **HoLot** chứa giá trị là 'Nguyen Van'

biến xâu **Ten** chứa giá trị là 'Hung'

phép ghép xâu

**HoLot** + ' ' + **Ten**

cho xâu kết quả là 'Nguyen Van Hung'

1. **Các phép so sánh**: bằng (=), khác (<>), nhỏ hơn (<), lớn hơn (>), nhỏ hơn hoặc bằng (<=), lớn hơn hoặc bằng (>=) thực hiện việc so sánh hai xâu theo qui tắc sau :

* Xâu A lớn hơn xâu B nếu như kí tự đầu tiên khác nhau giữa chúng kể từ trái sang trong xâu A có mã ASCII lớn hơn.

*Ví dụ* : 'Microsoft' > 'Microphone'

* Nếu A và B là các xâu có độ dài khác nhau và A là đoạn đầu của B thì A là nhỏ hơn B.

*Ví dụ* : 'May tinh' < 'May tinh cua toi'

* Hai xâu được coi là bằng nhau nếu như chúng giống nhau hoàn toàn. *Ví dụ* : Tin hoc' = Tin hoc'

1. **Các thủ tục và hàm chuẩn xử lí xâu**

• ***Delete(st, vt, n)*** thực hiện việc xóa ***n*** kí tự của biến xâu ***st*** bắt đầu từ vị trí ***vt***

*Ví dụ* :

|  |  |  |
| --- | --- | --- |
| **Giá trị** | **Thao tác** | **Kết quả** |
| st:='abcdef'; | Delete(st, 5, 2); | 'abcd' |
| st:= 'Song Hong'; | Delete(st, 1, 5); | 'Hong' |

• ***Insert(s1, s2, vt)*** chèn xâu ***s1*** vào xâu ***s2,*** bắt đầu ở vị trí ***vt.***

*Ví dụ* :

|  |  |  |
| --- | --- | --- |
| **Giá trị** | **Thao tác** | **Kết quả** |
| s1:=' PC '; s2:='IBM486'; | Insert(s1, s2, 4); | 'IBM PC 486' |
| S1:= '1'; s2:='Hinh .2'; | Insert(s1, s2, 6); | 'Hinh 1.2' |

• ***Copy(S, vt, N)*** tạo xâu gồm ***N*** kí tự liên tiếp bắt đầu từ vị trí ***vt*** của xâu ***S.***

*Ví dụ:*

|  |  |  |
| --- | --- | --- |
| **Giá trị** | **Thao tác** | **Kết quả** |
| s:='Bai hoc thu 9'; | Copy(s, 9, 5); | 'thu 9' |

*•* ***Length(s)*** cho giá trị là độ dài xâu ***s.***

*Ví dụ* :

|  |  |  |
| --- | --- | --- |
| **Giá trị** | **Thao tác** | **Kết quả** |
| s:='500 ki tu'; | Length(s); | 9 |

• ***Pos(s1, s2)*** cho vị trí xuất hiện đầu tiên của xâu ***s1*** trong xâu

***s2.***

*Ví dụ:*

|  |  |  |
| --- | --- | --- |
| **Giá trị** | **Thao tác** | **Kết quả** |
| s1:='cd'; s2:='abcdef'; | Pos(s1, s2); | 3 |
| s2:='abcdef; | Pos('k', s2); | 0 |

*•* ***Upcase(ch)*** cho chữ cái in hoa ứng với chữ cái trong ***ch.***

*Ví dụ* :

|  |  |  |
| --- | --- | --- |
| **Giá trị** | **Thao tác** | **Kết quả** |
| ch:='d'; | Upcase(ch); | 'D' |
| ch:='E'; | Upcase(ch); | 'E' |

**3. Một số ví *dụ****(tham khảo thêm SGK)*

xxxxx

**BÀI 13 : KIÊU BẢN GHI**(tham khảo thêm SGK)