

## 5. Câu lệnh rẽ nhánh và lặp

### a) Câu lệnh *case-of*

Xét bài toán: Lập chương trình nhập từ bàn phím tháng và năm rồi tính và đưa ra màn hình số ngày của tháng.

Câu lệnh *if-then* giải quyết việc rẽ nhánh tùy theo một trong hai khả năng. Bài toán trên liên quan tới việc chọn lựa một trong nhiều khả năng (rẽ nhiều nhánh). Khi đó, nếu dùng câu lệnh *if-then* giải quyết sẽ làm chương trình rườm rà. Câu lệnh chọn *case-of* trong TP cho phép rẽ nhiều nhánh một cách thuận lợi.

Câu lệnh *case-of* có dạng:

```
case <biểu thức nguyên hoặc kí tự> of  
    <danh sách 1>; <câu lệnh 1>;
```

< danh sách 2>: < câu lệnh 2>;

.....  
< danh sách N>: < câu lệnh N>

[ else < câu lệnh N + 1>]

end;

trong đó:

- Danh sách  $i$  ( $i = 1, 2, \dots, N$ ) là một hoặc nhiều giá trị của biểu thức nêu sau case, nếu có nhiều giá trị thì các giá trị viết cách nhau bởi dấu phẩy.
- Khi thực hiện, biểu thức sau từ khoá case sẽ được tính, giá trị nhận được sẽ lần lượt được kiểm tra xem nằm trong danh sách nào. Câu lệnh tương ứng với danh sách đầu tiên tìm thấy sẽ được thực hiện. Tiếp theo, thực hiện câu lệnh sau câu lệnh case-of.
- Nếu giá trị tính được không xuất hiện ở bất kì danh sách nào thì câu lệnh N + 1 sau else (nếu có) sẽ được thực hiện.

Chương trình để giải bài toán vừa nêu như sau:

```
program Vi_du_case;
uses crt;
var T, N, SN: integer;
begin
  clrscr;
  write('Cho biet thang va nam: ');
  readln(T,N);
  case T of
    4,6,9,11: SN:= 30;
    2: if (N mod 400=0) or ((N mod 100<>0) and (N mod 4=0))
        then SN:= 29 else SN:= 28
    else SN:= 31
  end;
  writeln('Thang ', T, ' nam ', N, ' co ', SN, ' ngay. ');
  readln
end.
```

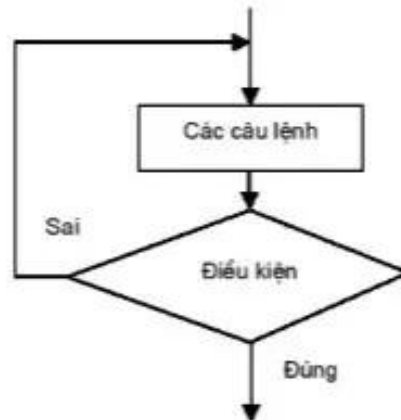
#### b) Câu lệnh lặp repeat-until

Câu lệnh này dùng để tổ chức lặp với số lần lặp không biết trước. Khác với câu lệnh while-do, điều kiện kết thúc lặp trong câu lệnh repeat-until được kiểm tra sau và có dạng:

```
repeat
  <dãy câu lệnh>;
until <điều kiện>;
```

trong đó điều kiện là biểu thức quan hệ hoặc logic.

Việc thực hiện lệnh *repeat-until* được thể hiện trên sơ đồ sau.



Hình P.8. Sơ đồ thực hiện lệnh *repeat-until*

### Ví dụ 1

Xét đoạn chương trình:

```
repeat  
  write('HAY NHAP SO NGUYEN N: ');  
  readln(N)  
until (N>0) and (N <= 100);
```

Đoạn chương trình này dùng để nhập số nguyên  $N$  thỏa mãn điều kiện  $0 < N \leq 100$ . Nếu giá trị nguyên đưa vào nằm ngoài khoảng này thì chương trình sẽ yêu cầu nhập lại cho đến khi giá trị nhập vào thỏa mãn điều kiện.

### Ví dụ 2

Tính gần đúng  $\sqrt{A}$  ( $A > 0$ ) chính xác đến chữ số thập phân thứ năm theo phương pháp lặp Niu-tơn.

Theo phương pháp này, ta cần tính dãy  $X_0, X_1, X_2, \dots$ , theo công thức:

$$X_0 = 1$$

...

$$X_{n+1} = \frac{1}{2} \left( X_n + \frac{A}{X_n} \right), \quad n = 0, 1, 2, \dots \quad (*)$$

Quá trình lặp kết thúc khi  $|X_{n+1} - X_n| < 0,00001$ . Giá trị của  $X_{n+1}$  được lấy làm giá trị gần đúng của  $\sqrt{A}$ .

**Input:** Giá trị thực  $A$ , nhập vào từ bàn phím.

**Output:** Đưa ra màn hình giá trị gần đúng của  $\sqrt{A}$ .

Ta chỉ cần dùng hai biến thực trung gian  $XNEW$  và  $XOLD$  để chứa tương ứng các giá trị  $X_{n+1}$  và  $X_n$ .

Thuật toán được mô tả như sau:

Bước 1. Nhập A;

Bước 2.  $X_{NEW} \leftarrow 1$  {giá trị  $X_0$ };

Bước 3.  $X_{OLD} \leftarrow X_{NEW}$  {chuyển giá trị mới thành giá trị cũ};

Bước 4.  $X_{NEW} \leftarrow \frac{1}{2} \left( X_{OLD} + \frac{A}{X_{OLD}} \right)$  {Tính lại giá trị mới theo  $(*)$ };

Bước 5. Nếu  $|X_{NEW} - X_{OLD}| \geq 0,00001$  thì quay về bước 3;

Bước 6. Đưa giá trị  $X_{NEW}$  ra màn hình, rồi kết thúc.

Chương trình thể hiện thuật toán trên như sau:

```
program CAN_BAC_2;
uses crt;
var A, XNEW, XOLD: real;
begin
  clrscr;
  repeat
    write(' A = '); readln(A);      {Buoc 1}
  until A > 0;
  XNEW:= 1;      {Buoc 2}
  repeat
    XOLD:= XNEW;      {Buoc 3}
    XNEW:= (XOLD + A/XOLD)/2;      {Buoc 4}
  until ABS(XNEW - XOLD) < 0.00001; {Buoc 5}
  writeln('Can bac 2 cua ',A:10:5, ' la ',XNEW:10:5);{Buoc 6}
  readln
end.
```