Bài thực hành 5

SỬ DỤNG LỆNH LẶP FOR...DO

1. Mục đích, yêu cầu

- Viết chương trình Pascal có câu lệnh lặp `for...do`.
- Tiếp tục nâng cao kĩ năng đọc hiểu chương trình.

2. Nội dung

BÀI 1. Viết chương trình in ra màn hình bảng nhân của một số từ 1 đến 9, số được nhập từ bàn phím và dừng màn hình để có thể quan sát kết quả.

```
Nhap so N=8

Bang nhan 8

8 x  1 =  8
8 x  2 = 16
8 x  3 = 24
8 x  4 = 32
8 x  5 = 40
8 x  6 = 48
8 x  7 = 56
8 x  8 = 64
8 x  9 = 72
8 x 10 = 80
```

Hình 36

a) Gõ chương trình sau đây:

```
uses crt;
var N, i: integer;
begin
    clrscr;
    write('Nhap so N ='); readln(N);
    writeln;
    writeln('Bang nhan ',N);
    writeln;
    for i:= 1 to 10 do writeln(N,' x ', i:2,' = ', N*i:3);
    readln
end.
```

b) Tìm hiểu ý nghĩa của các câu lệnh trong chương trình, dịch chương trình và sửa lỗi, nếu có.

c) Chạy chương trình với các giá trị nhập vào lần lượt bằng 1, 2,..., 10. Quan sát kết quả nhận được trên màn hình.

BÀI 2. Chỉnh sửa chương trình để làm đẹp kết quả trên màn hình.

Kết quả của chương trình nhận được trong bài 1 có hai nhược điểm sau đây:

- Các hàng kết quả quá sát nhau nên khó đọc;
- Các hàng kết quả không được cân đối với hàng tiêu đề.

Nên sửa chương trình bằng cách chèn thêm một hàng trống giữa các hàng kết quả và đẩy các hàng này sang phải một khoảng cách nào đó.

Hình 37

a) Chỉnh sửa câu lệnh lặp của chương trình như sau:

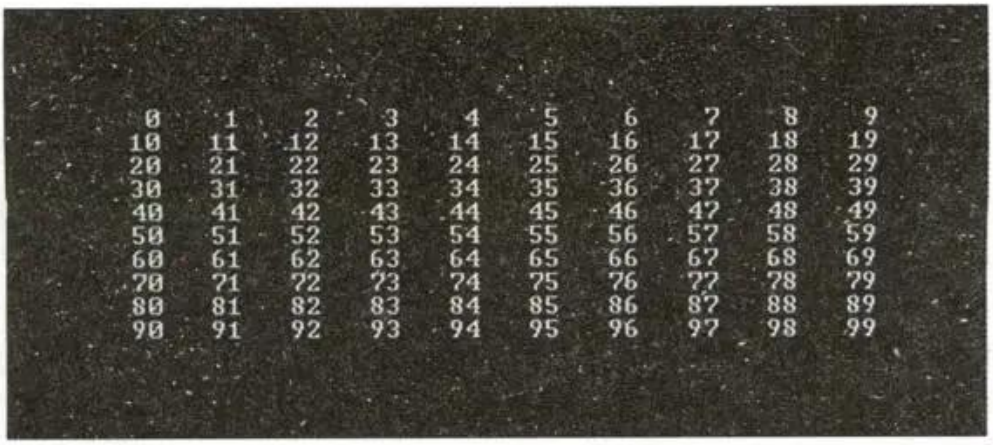
```
for i:= 1 to 10 do
begin
    GotoXY(5,WhereY); writeln(N,' x ', i:2,' = ', N*i:3);
    writeln
end;
```

Lưu ý:

- Chỉ sử dụng được các lệnh `GotoXY`, `WhereX` và `WhereY` sau khi khai báo thư viện `crt` của Pascal.
- Màn hình máy tính được chia thành các cột và các hàng, được tính bắt đầu từ góc trên bên trái. Câu lệnh `GotoXY(a,b)` có tác dụng đưa con trỏ về *cột a, hàng b*.
- `WhereX` cho biết số thứ tự của *cột* và `WhereY` cho biết số thứ tự của *hàng* đang có con trỏ. Ví dụ `GotoXY(5,WhereY)` đưa con trỏ về vị trí cột 5 của hàng hiện tại.

b) Dịch và chạy chương trình với các giá trị gõ vào từ bàn phím. Quan sát kết quả nhận được trên màn hình.

BÀI 3. Cũng như câu lệnh `if`, có thể dùng câu lệnh `for` lồng trong một câu lệnh `for` khác khi thực hiện lặp. Sử dụng các câu lệnh `for...do` lồng nhau để in ra màn hình các số từ 0 đến 99 theo dạng bảng như hình sau:



Hình 38

a) Tìm hiểu chương trình sau đây :

```
program Tao_bang;
uses crt;
var
i: byte; {chi so cua hang}
j: byte; {chi so cua cot}
begin
clrscr; {xoa man hinh}
for i:= 0 to 9 do  {viet theo tung hang}
   begin
        for j:= 0 to 9 do {viet theo tung cot tren moi hang}
        write(10*i+j :4); {viet cac so ij ra man hinh}
        writeln;   {xuong hang moi}
   end;  {xong hang thu i}
readln {dung chuong trinh de xem ket qua}
end.
```

b) Gõ và chạy chương trình, quan sát kết quả trên màn hình. Sử dụng thêm các câu lệnh `GotoXY(a,b)` để điều chỉnh (một cách tương đối) bảng kết quả ra giữa màn hình.

TỔNG KẾT

1. Cấu trúc lặp với số lần lặp biết trước được thể hiện bằng câu lệnh Pascal **for...do**.
2. Giống như các câu lệnh rẽ nhánh **if...then**, các câu lệnh **for...do** cũng có thể lồng trong nhau. Khi đó các *biến đếm* trong các câu lệnh lặp phải khác nhau.
3. Câu lệnh `GotoXY(a,b)` có tác dụng đưa con trỏ về *cột a, hàng b*. `WhereX` cho biết số thứ tự của *cột* và `WhereY` cho biết số thứ tự của hàng đang có con trỏ.
4. Có thể kết hợp câu lệnh `GotoXY(a,b)` với các hàm chuẩn `WhereX` và `WhereY` để điều khiển vị trí của con trỏ trên màn hình.

Đọc thêm 1. Tính gần đúng số π

Như đã biết, số π đầu tiên được dùng để tính chu vi và diện tích hình tròn. Sau này số π còn được sử dụng trong nhiều tính toán khác nữa. π là số vô tỉ nên không thể biểu diễn chính xác bằng các số thập phân. Các nhà toán học đã có rất nhiều cố gắng để tính xấp xỉ số π .

Có rất nhiều công thức để tính xấp xỉ số π. Sau đây là một công thức đơn giản:

Công thức. Số π có thể tính xấp xỉ bằng công thức sau đây:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots + \frac{1}{2n-1} - \frac{1}{2n+1} + \ldots$$

Điều có vẻ như khó thể hiện trong lệnh lặp là các số hạng bên phải của tổng luân phiên đổi dấu. Nếu như tất cả các số hạng đều có dấu dương thì là một vòng lặp đơn giản. Vì vậy, thuật toán đã sử dụng một biến phụ, đặt giá trị ban đầu bằng 1 và nhân với nhau trong mỗi vòng lặp để giải quyết vấn đề này.

Dưới đây là thuật toán tính số π theo công thức trên với giá trị *n* được cho từ bàn phím:

Bước 1. Nhập *n*.

Bước 2. Pi ← 0, *i* ← 0, *dau* ← 1.

Bước 3. *Pi* ← *Pi* + *dau* ×1/(2*i* + 1), *dau* ← − *dau*.

Bước 4. *i* ← *i* + 1. Nếu *i* ≤ *n*, quay trở lại bước 3.

Bước 5. Đưa ra kết quả 4 × *Pi* và kết thúc.

Hãy kiểm tra thuật toán với các giá trị *i* = 0, 1,..., 5 để đảm bảo công thức đúng.

Sau đây là chương trình tính xấp xỉ số π:

```
program So_Pi;
var SoPi: real;
    i, dau: integer;
    n: longint;
begin
    write('Hay cho so cac so hang de tinh so Pi : ');
    readln(n); {so cac so hang cua tong}
    if n > 0 then
    begin
     SoPi:= 0; {dat cac gia tri khoi dau}
     dau:= 1; {dung de doi dau}
     for i:= 0 to n do
      begin SoPi:= SoPi + dau*1/(2*i + 1);
            dau:= -dau end;
     writeln('So Pi gan bang ', SoPi*4);
    end  {khong co dau; vi dung truoc else}
    else
    begin
    writeln('So ban nhap phai > 0');
    writeln('Nhan phim Enter va chay lai chuong trinh');
    end;
    readln
end.
```

Hãy tìm hiểu, gõ, dịch và chạy chương trình với các giá trị lần lượt n = 1000, 10000, 12000, 15000 và quan sát kết quả của chương trình.

Nhận xét. Đây chỉ là một công thức để viết chương trình tính số π một cách đơn giản. Để tính xấp xỉ số π với hàng triệu, hàng tỉ số lẻ, người ta phải sử dụng các công thức và thuật toán khác hiệu quả hơn.

Pascal cũng có hàm chuẩn *Pi* để cho giá trị của số π .