



Bài thực hành 6

SỬ DỤNG LỆNH LẬP WHILE...DO

Thời lượng: 2 tiết

1. Mục đích, yêu cầu

- Hiểu câu lệnh lập *while...do* trong chương trình;
- Biết lựa chọn câu lệnh lập *while...do* hoặc *for...do* phù hợp với tình huống cụ thể;
- Rèn luyện kỹ năng về khai báo, sử dụng biến;
- Rèn luyện khả năng đọc chương trình;
- Biết vai trò của việc kết hợp các cấu trúc điều khiển.

2. Những điểm cần lưu ý và gợi ý dạy học

a) Trước hết cần lưu ý rằng chuẩn kiến thức, kỹ năng không yêu cầu HS phải viết được chương trình có sử dụng câu lệnh lập với số lần chưa biết trước. Do vậy, trong bài thực hành này không yêu cầu HS phải tự viết chương trình có câu lệnh *while...do* để giải bài toán. GV có thể cho HS đọc, hiểu, thử chạy chương trình trên máy và đặc biệt cần hiểu được hoạt động của lệnh *while...do* trong chương trình.

b) Với bài 1, trước hết cần xác định Input và Output của bài toán:

Input: Dãy số thực x_1, x_2, \dots, x_n ;

Output: Giá trị trung bình $(x_1 + x_2 + \dots + x_n)/n$.

Thuật toán

Bước 1. Nhập n là số lượng các số thực sẽ được nhập từ bàn phím:

1.1. $Dem \leftarrow 0$;

1.2. $Sum \leftarrow 0$.

Bước 2. Trong khi $Dem < N$ thì

2.1. $Dem \leftarrow Dem + 1$;

2.2. Nhập giá trị số thực x từ bàn phím;

2.3. $Sum \leftarrow Sum + x$;

Bước 3. $TB \leftarrow Sum/N$.

Bước 4. Đưa giá trị TB ra màn hình, rồi kết thúc.

Có nhiều cách để mô tả thuật toán này, tuy nhiên cách mô tả trên đây được sử dụng với mục đích HS thuận lợi hơn khi đọc, hiểu, đối chiếu giữa thuật toán với chương trình *Tinh_Trung_binh* ở câu b.

Căn cứ vào mô tả thuật toán, HS tìm hiểu để xác định các biến và kiểu dữ liệu tương ứng cần khai báo trong chương trình.

Câu c yêu cầu HS dịch, chỉnh sửa, chạy và kiểm thử chương trình. Những kỹ năng này HS đã được rèn luyện ở những bài thực hành trước cho nên HS có thể hoàn toàn thực hiện được. GV có thể đưa ra hoặc hướng dẫn HS tạo ra những bộ dữ liệu test.

GV cần yêu cầu HS đọc, thảo luận đối chiếu giữa thuật toán và các câu lệnh mô tả thuật toán trong chương trình. Cần làm cho HS hiểu rõ về hoạt động của lệnh lặp *while...do* trong chương trình, có thể cho HS làm việc nhóm để mô phỏng chương trình (việc này nên được làm trên lớp học, ở tiết bài tập trước khi thực hành trên máy).

Ví dụ dưới đây là một mô phỏng hoạt động chính của chương trình với $n = 3$:

1. Trước khi bắt đầu vòng lặp *while...do*: $dem = 0, TB = 0, n = 3$;
2. Bắt đầu lệnh lặp *while...do*

Giá trị DEM trước khi thực hiện vòng lặp	DEM < n	Giá trị DEM sau khi thực hiện vòng lặp	X (nhập từ bàn phím)	TB
0	Đúng	1	10	10
1	Đúng	2	15	25
2	Đúng	3	20	45
3	Sai			

3. Kết thúc lệnh lặp *while...do*: $TB = 45/3 = 15$.

Câu d yêu cầu HS chuyển từ sử dụng câu lệnh *while...do* sang sử dụng câu lệnh *for...do*. Qua việc làm này HS được rèn luyện thêm về sử dụng lệnh *for...do*.

Tuy nhiên, về cơ bản tình huống sử dụng *while...do* và *for...do* là khác nhau. *While...do* thích hợp hơn với trường hợp lặp với số lần chưa biết trước, *for...do* thích hợp hơn với trường hợp lặp với số lần biết trước.

Như vậy, ở câu d, bên cạnh mục đích cho HS có ý thức trong việc lựa chọn cấu trúc lặp phù hợp với tình huống, còn có mục đích tiếp tục rèn luyện viết chương trình với câu lệnh *for...do* đảm bảo đạt yêu cầu đề ra trong chuẩn kiến thức, kỹ năng.

c) Với bài 2, cách tiến hành giống như với bài 1. Trước hết cần xác định Input và Output của bài toán:

Input: Số tự nhiên N ;

Output: Trả lời N là số nguyên tố hoặc N không là số nguyên tố.

Thuật toán

HS lớp 8 đã biết tính chất của số nguyên tố là số tự nhiên chỉ chia hết cho 1 và chính nó.

Để kiểm tra N có phải số nguyên tố hay không ta kiểm tra xem N có chia hết cho các số từ 2 đến $N - 1$ hay không. Nếu N không chia hết cho số nào trong khoảng từ 2 đến $N - 1$ thì N là số nguyên tố, ngược lại nếu N chia hết cho bất kì một số nào trong khoảng từ 2 đến $N - 1$ thì N không phải là số nguyên tố.

Sử dụng phép chia lấy phần dư *mod* để kiểm tra tính chia hết.

Bước 1. Nhập số tự nhiên N từ bàn phím.

Bước 2. Nếu $N \leq 1$ thông báo N không phải là số nguyên tố, rồi chuyển đến bước 4.

Bước 3. Nếu $N > 1$:

3.1. $i \leftarrow 2$;

3.2. Trong khi $N \bmod i \neq 0$ còn đúng thì $i \leftarrow i + 1$;

3.3. Nếu $i = N$ thì thông báo N là số nguyên tố, rồi chuyển đến bước 4.

Ngược lại, thông báo N không phải là số nguyên tố;

Bước 4. Kết thúc.

Sau đó GV cho HS đọc chương trình trong SGK, đối chiếu việc sử dụng câu lệnh để mô tả thuật toán trên đây.

Lưu ý: Trong chương trình sử dụng cả câu lệnh điều kiện, câu lệnh lặp *while...do*. Mục đích của bài này là để HS thấy được sự cần thiết phải kết hợp

các cấu trúc điều khiển để giải quyết bài toán. Hơn nữa, trong ví dụ này còn sử dụng phép chia lấy phần dư *mod*. Điều này thể hiện sự cần thiết và tính hiệu quả khi lựa chọn công cụ phù hợp trong lập trình.

d) Đối với đa số HS lớp 8, thuật toán kiểm tra tính nguyên tố của một số tự nhiên là không khó. Tuy nhiên, nếu thấy HS của mình có thể gặp khó khăn khi tìm hiểu thuật toán này, GV có thể lấy một số giá trị *n* cụ thể để "chạy" mô phỏng hoạt động của chương trình hoặc thay thế bằng ví dụ khác. Ví dụ mà GV đưa ra có thể chỉ cần thể hiện sự kết hợp giữa câu lệnh điều kiện và câu lệnh lặp với số lần chưa biết trước, không nhất thiết phải có tình huống sử dụng phép chia lấy phần dư *mod*.

Sự kết hợp các cấu trúc điều khiển (tuần tự, rẽ nhánh và lặp) trong ngôn ngữ lập trình tạo nên sự linh hoạt và góp phần tạo nên sức mạnh của ngôn ngữ lập trình. Chính sự kết hợp giữa các cấu trúc điều khiển cho phép ngôn ngữ lập trình mô tả được những thuật toán phức tạp, giúp giải quyết được nhiều bài toán xuất phát từ nhu cầu thực tiễn. GV có thể không cần trình bày ý nghĩa của việc kết hợp các cấu trúc điều khiển với HS.

Như trên đã nêu chuẩn kiến thức, kỹ năng không yêu cầu HS phải viết được chương trình có sử dụng câu lệnh lặp với số lần biết trước. Tuy nhiên, nếu HS tiếp thu tốt, GV có thể yêu cầu HS tập viết chương trình đơn giản có sử dụng câu lệnh *while...do*.