

Câu 1: Viết một biểu thức mô tả các tên file trong hệ thống DOS cũ, biết rằng nó bao mỗi chuỗi có đúng 8 kí tự bao gồm chữ cái hoặc dấu gạch dưới và bắt đầu bằng chuỗi abcde.
Ví dụ: abcdeCSE, abcdeLOL

Câu 2: Ngôn ngữ lập trình Y bao gồm các khai báo biến như sau:

- Một khai báo biến bao gồm từ khóa **let** (LET) và tiếp theo là một danh sách các biến được cách nhau bởi một dấu phẩy (,) (CM). Tiếp theo là phần khai báo kiểu của biến bắt đầu bởi một từ là 3 dấu chấm (...) (TDOTS) và đi theo sau một danh sách các kiểu. Các kiểu có thể dùng để khai báo bao gồm **int**, **real**, **string**, **list**.

- Mỗi biến có thể được khởi tạo hoặc không được khởi tạo, nếu có, thì trong danh sách khai báo biến sẽ có phần khởi tạo. Phần khởi tạo bắt đầu bằng một dấu bằng (=) (EQ) và theo sau là danh sách các biểu thức tương ứng với thứ tự các biến được nêu trước đó. Nếu biến nào không khởi tạo, sử dụng kí tự gạch dưới (_) (US) để thay thế.

- Một biểu thức là tổ hợp giữa các toán tử và toán hạng. Trong Y, phép toán * (MUL), / (DIV) có độ phức tạp cao nhất và có tính kết hợp phải, trong đó + (PLUS), - (MINUS) lần lượt có độ ưu tiên trung bình và thấp với + có tính kết hợp trái còn - thì không có tính kết hợp. Để thay đổi sự ưu tiên và tính kết hợp, người ta dùng biểu thức con, bao bởi cặp ngoặc tròn () (LR – RR). Các toán tử có thể là số nguyên (INTLIT), số thực (REALLIT), chuỗi (STRINGLIT) và danh hiệu (ID).

Ví dụ: let a, b, c ... int, real, string = 1 + 2 * 3, _, 15 + x;

Câu 3: Thế nào là tính đa hình (polymorphism) trong lập trình hướng đối tượng? Bằng kiến thức các anh/ chị đã học và một ngôn ngữ lập trình quen thuộc, hãy nêu ví dụ về tính đa hình và sự khác biệt về tính đa hình trong ngôn ngữ anh/ chị đã lấy ví dụ?

Câu 4:

a. Hãy viết hàm length(lst) nhận vào một danh sách và trả về độ dài của danh sách mà không được phép sử dụng hàm len() của Python. Gợi ý: sinh viên phải sử dụng hàm map và hàm sum.

b. Hãy viết hàm iter(f, n) trả về một hàm tính gộp đầu n lần với giá trị nhận vào.

Ví dụ: f = lambda x: x * 2

g = iter(f, 3)

u = g(3) # u = (((3 * 2) * 2) * 2) = 64