

**ĐỀ KIỂM TRA: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT 1**  
**HỌC KỲ 1, NĂM HỌC: 2021-2022      THỜI GIAN: 70 PHÚT**  
**LỚP: TNK44**

(Sinh viên được sử dụng tài liệu)

Viết 1 chương trình hoàn chỉnh thực hiện các chức năng sau trên danh sách liên kết đơn với dữ liệu là các số nguyên:

- 1) (a) Khai báo cấu trúc dữ liệu của danh sách liên kết đơn; (b) viết hàm *NodePointer TaoNut(X)* để tạo một nút mới có trường dữ liệu là một số nguyên  $X$  cho trước; (c) Viết hàm chèn đầu *void ChenDau(List, X)* để chèn nút có dữ liệu  $X$  vào đầu danh sách *List*.
- 2) (a) Viết hàm *LL TaoDS()* để tạo một danh sách với dữ liệu được nhập từ bàn phím (sử dụng hàm *ChenDau* ở câu 2); và (b) Viết hàm *void SuaDS(List, X, Y)* để sửa tất cả nút trong danh sách *List* có dữ liệu  $X$  thành dữ liệu  $Y$ . Ví dụ, cho  $List = [2, 3, 5, 3, 7, 9]$ ,  $X = 3$  và  $Y = 8$ , sau khi sửa danh sách sẽ là:  $List = [2, 8, 5, 8, 7, 9]$ .
- 3) Viết hàm *void TachDS(List1, List2, List3, X)* để tách danh sách *List1* thành 2 danh sách *List2* và *List3* theo yêu cầu sau:
  - *List2* chứa tất cả các nút trong *List1* có dữ liệu lớn hơn  $X$ , và *List3* chứa các nút trong *List1* có dữ liệu nhỏ hơn hoặc bằng  $X$ .
  - Các nút trong *List2* và *List3* có thứ tự ngược với thứ tự mà chúng xuất hiện trong *List1*.Ví dụ, cho  $List1 = [3, 9, 2, 7, 8, 4]$  và  $X = 6$ , sau khi tách  $List2 = [8, 7, 9]$  và  $List3 = [4, 2, 3]$ .
- 4) Viết hàm xóa nút cuối cùng có dữ liệu là  $X$ .
- 5) Viết hàm *main* minh họa tất cả các hàm đã viết ở trên.

-----HẾT-----