Họ và tên : Tống Anh Kiệt   
 MSSV : N22DCDK039  
Lớp : D22CQDK01-N|

Đề số 3:

BÀI 1

Câu A : Giả sử ta cần tính toán trên các số có dạng a + b√7 với a, b là các số nguyên. Các

thao tác trên kiểu số này bao gồm nhập, xuất, phép cộng, phép trừ, phép đổi dấu,

phép nhân.  
Bài làm : <Bài-1(a).cpp>  
  
Câu B : b) Hãy cài đặt mã cho các thao tác kể trên.  
Bài làm : <Bài-1(b).cpp>  
  
BÀI 2

Trình bày và cho ví dụ mã nguồn về hàm hủy ảo (virtual destructor) trong ngôn

ngữ lập trình C++. Khi lập trình với ngôn ngữ lập trình C++, nếu không dùng hàm

hủy ảo thì có nguy cơ gì? Bạn hãy cho ví dụ cụ thể có minh họa bằng mã nguồn.  
  
Hàm hủy ảo (virtual destructor) trong C++:

Hàm hủy ảo là một hàm hủy được đánh dấu bằng từ khóa virtual. Khi một lớp có một hàm hủy ảo và một lớp kế thừa từ lớp đó, nếu đối tượng của lớp kế thừa được giải phóng (delete) thông qua con trỏ của lớp cơ sở, hàm hủy ảo của lớp kế thừa sẽ được gọi.Hàm hủy ảo (virtual destructor) trong C++:  
Vd : <Bài-2(a).cpp>  
  
Nguy cơ khi không sử dụng hàm hủy ảo không sử dụng hàm hủy ảo khi có đa hình (polymorphism) và giải phóng đối tượng thông qua con trỏ của lớp cơ sở, hàm hủy của lớp con không được gọi, dẫn đến rủi ro rò rỉ bộ nhớ.  
vd : <Bài-2(b).cpp>  
  
BÀI 3

Một cấu trúc dữ liệu hướng đối tượng được cài đặt bằng mã nguồn C++ như trong

bảng sau. Để làm bài này, bạn cần đọc kỹ mã nguồn trước khi thực hiện các câu

theo yêu cầu của đề bài.  
a) Có thể tạo ra đối tượng của lớp absList được hay không? Lớp absList được gọi

là lớp gì?  
Không thể lớp absList được. Lý Do: Vì đây là lớp trừu tượng với dấu hiệu addFirst(int pId), addFirst(int pId), showAll(ostream&) là phương thức thuần ảo.  
Lớp absList được gọi là abstract Class tức là lớp trừu tượng  
b) Dòng lệnh đánh nhãn số 6 (nghĩa là lệnh delete lnkLst) làm cho hàm hủy của

lớp linearList chạy bao nhiêu lần? Hãy giải thích.  
 Số lần hủy: Hủy 9 lần.   
 Giải thích: Vì trong mỗi lần phương thức addFirst(int) hoạt động thì lnkLst trỏ vào một con trỏ mới và từ con trỏ đầu tiên ta truy xuất vào thành phần bên trong. Về mặt khác ta có thể nói trong lớp linearList absList\* subLst chính là pNext của danh sách liên kết tự định nghĩa.  
  
c) Cài đặt lại mã nguồn của phương thức countAll() trong lớp linearList:  
bài làm : <Bài-3(c).cpp>  
  
d) Phương thức showAll(ostream& outDev) của lớp linearList chưa cài đặt xong.

Bạn hãy cài đặt mã nguồn của phương thức này.  
bài làm : <Bài-3(d).cpp>  
  
e) Cấu trúc dữ liệu được cài đặt bởi lớp linearList là một danh sách liên kết đơn (singly linked list). Mỗi đối tượng linearList chứa một phần tử dữ liệu và một con trỏ đến đối tượng linearList tiếp theo trong danh sách. Nhận xét thêm có thể là đây là một cách cài đặt đơn giản của danh sách liên kết đơn, nhưng còn thiếu một số tính năng, chẳng hạn như hàm hủy của lớp linearList chỉ giải phóng bộ nhớ của phần tử đầu tiên, còn phần tử con trỏ không được giải phóng