Câu 2

Hàm hủy ảo (virtual destructor) trong C++ là một khái niệm quan trọng khi làm việc với kế thừa và polymorphism. Khi bạn có một lớp cón sở và lớp dẫn xuất từ lớp cơ sở, việc sử dụng hàm hủy ảo giúp đảm bảo rằng các hàm hủy của lớp dẫn xuất được gọi theo thứ tự, ngay cả khi bạn truy cập đối tượng dựa trên con trở của lớp cơ sở.

VD

#include<iostream>

using namespace std;

class Base{

public:

Base(){

cout << “Base constructor”<< endl;

}

Virttual ~Base{

cout << “Base destructor”<< endl;

}

};

Class Derived : public Base{

public:

Derived(){

cout << “ Derived constructor”<< endl;

}

~Derived(){

cout << “ Derived destructor”<< endl;

}

};

int main()

{

Base \*obj=new Derived();

delete obj;

return 0;

}

**Câu 3**

a)

Không thể khởi tạo lớp absList vì đây là lớp trừu tượng có các phương thức thuần ảo

Lớp absList là lớp trừu tượng

b)

Số lần hủy: Hủy 9 lần.

Giải thích: Vì trong mỗi lần phương thức addFirst(int) hoạt động thì InkLst 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\* subLst chính là pNext của danh sách liên kết tự định nghĩa.

e)

Cấu trúc dữ liệu: Ngăn xếp stack